

Reference

NBS
PUBLICATIONS

NBSIR 84-3009

BUILDING PENETRATION PROJECT

NAT'L INST. OF STAND & TECH



A11106 033823

National Bureau of Standards
U.S. Department of Commerce
Boulder, Colorado 80303

September 1984

QC

100

.U56

84-3009

1984

NBSIR 84-3009

BUILDING PENETRATION PROJECT

Ref
AC100
- U56
NBSIR 84-3009
1984

J.C. Wyss
W.J. Anson
R.D. Orr

Electromagnetic Fields Division
National Engineering Laboratory
National Bureau of Standards
U.S. Department of Commerce
Boulder, Colorado 80303

September 1984

Prepared for:
Communications Electronics Engineering Installation Agency
Fort Huachuca, Arizona 08613



U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

CONTENTS

Page

ACKNOWLEDGMENTS.....	iv
1. INTRODUCTION.....	1
1.1 Background.....	1
1.2 The Approach.....	1
1.3 Outline.....	2
2. SEARCH FOR AND COMPUTATION OF ELECTROMAGNETIC PROPERTIES OF BUILDING MATERIALS.....	3
2.1 Introduction.....	3
2.2 The Literature Search.....	3
2.3 Computations.....	4
2.4 Uncertainty.....	10
2.5 Reflection Coefficient and Attenuation Data for Selected Building Materials.....	13
3. CALCULATION FOR BUILDING SHIELDING EFFECTIVENESS -- THEORY.....	19
3.1 Introduction.....	19
3.2 Definition of Problem.....	19
3.3 Input Data Available.....	19
3.4 Approach.....	20
3.5 Input Resonances.....	23
4. COMPUTER PROGRAM DESCRIPTIONS.....	27
4.1 Introduction.....	27
4.2 General Description of the Field Calculation Program.....	27
4.3 General Description of the Data Entry Programs.....	28
4.4 Data Structures Used in the Programs.....	28
4.5 Description of Programs and Subroutines Used in Field Calculation.....	32
5. USERS' GUIDE TO COMPUTER PROGRAMS FOR DATA ENTRY AND COMPUTATION.....	37
5.1 Introduction.....	37
5.2 Data Preparation for Programs SWALLS, STYPES, AND SHOLES.....	38
5.3 How to Become a Remote-Site Time-Share User.....	40
5.4 The CDC 170/750:Data Entry and Computation.....	41
5.4.1 Log-In Procedure.....	41
5.4.2 Procedure File MSTORE.....	41
5.4.3 Data File BxxxxxW (W for Walls).....	42
5.4.4 Data File BxxxxxT (T for Types).....	43
5.4.5 Data File BxxxxxH (H for Holes).....	43
5.4.6 Data File BxxxxxF (F for Frequency).....	44
5.4.7 Computation Program MASTER.....	44
6. EXPERIMENTAL DATA.....	58
6.1 Sierra and Seneca Falls Army Depots.....	58
6.2 Naval Training Equipment Center (NTEC).....	59
7. COMPUTER RESULTS AND CONCLUSIONS.....	75
7.1 Comparison With Experimental Results.....	75
7.2 Recommendation for Further Work.....	76
8. REFERENCES.....	85
9. APPENDICES.....	86
9.1 Test Plan for Building Attenuation Measurements.....	86
9.2 Listing of Computer Program SMATDB.....	90
9.3 Listing of Computer Program SHOLES.....	109
9.4 Listing of Computer Program SWALLS.....	132
9.5 Listing of Computer Program STYPES.....	161
9.6 Listing of Computer Program SFREQ.....	187
9.7 Listing of Computer Program MASTER.....	207
9.8 Blank Forms for Data Taking.....	302

ACKNOWLEDGMENTS

We would like to thank E. M. Livingston for her excellent help in performing the library searches that provided much of the data used in this project. Besides thanking J. E. Cruz and G. R. Reeve for supervising and taking the experimental data, we would also like to thank the staff at Fort Huachuca for their help in making the field measurements. Stephen Sheeran was very helpful in writing some of the printout routines, in running some of the test cases, and in graphing and drawing many of the figures. We also acknowledge the financial support from the Communications Electronics Engineering Installation Agency, Fort Huachuca, that made this study possible.

it penetrates the floor, ceiling, and each wall. The total radiant energy within a given room is then the sum of the energies arriving there via the various room-to-room routes within the building. The ratio (in dB) of this total radiant energy density to the energy density incident upon the building is the attenuation, or shielding effectiveness, assigned to that room.

1.3 Outline

Chapter two reviews the search for electrical properties of common building materials, and the mathematical expressions used to compute wall attenuation from these properties. We list the several computer data bases consulted, and note the key word groups that summoned the most useful references. Brief derivations of the computation expressions are given.

In chapter three we present the theory supporting our procedure for computing the attenuation of electromagnetic signals by buildings. It is here that we discuss the assumptions in the formulation of the problem, and limitations imposed by those assumptions.

Descriptions of the data entry programs, data files, and the computation program MASTER constitute chapter four. This material documents the programs and will be of interest mainly to the person wishing to know more about their workings. (User instructions are in the User's Guide, chapter five.)

Chapter five is a guide to the use of the programs. There are instructions on how to become a time-share user, and how to organize data and enter it into the data files. Sample data tabulations are given; also examples of computer printouts illustrating user-computer conversations during data entry.

In chapter six are the results of building attenuation measurements made by NBS at three Army installations. Graphs present the measured data (in dB) versus frequency, and show the effect of the direction of incidence (i.e., location of the launching antenna) on the shielding effectiveness of the structures evaluated. Building floor plans show measurement locations and the placement of launching antennas.

Conclusions and bibliography are chapters seven and eight, respectively. Appendix 9.1 details procedures for making building attenuation measurements and assigning estimated uncertainty. Appendices 9.2-9.7 contains the listings for the five data entry programs, and for the computation program MASTER. Appendix 9.8 is a set of copyable forms for tabulating data to be entered into the data files.

2. SEARCH FOR AND COMPUTATION OF ELECTROMAGNETIC PROPERTIES OF BUILDING MATERIALS

2.1 Introduction

This chapter begins with a description of the literature search for data on the shielding effectiveness of building materials and of buildings themselves. The search was primarily a computer interrogation of several large data bases, although we also perused locally available journals, handbooks, reports and conference proceedings. This extensive search yielded only a few useful references, just one of which furnished us with most of our experimental data on conventional building materials.

Our computations of reflection coefficient and attenuation are based on expressions derived in most introductory texts on electromagnetic fields or electricity and magnetism. We briefly discuss these expressions and their use in obtaining the data in thirteen tables at the end of this chapter.

2.2 The Literature Search

To carry out the computer literature search, we relied on the expertise of Mrs. Victoria R. Schneller of Library Services, Environmental Research Laboratories, National Oceanic and Atmospheric Administration, Boulder, Colorado. The data bases consulted were:

- NTISearch, the computer search service of the National Technical Information Service (NTIS) of the U.S. Dept. of Commerce; accesses all technical abstracts compiled by NTIS; 1964 - present.
- INSPEC (Information Service for Physics, Electrotechnology, and Control); comprises Physics Abstracts, Electrical and Electronic Abstracts, and Computer and Control Abstracts; established by the Institute of Electrical Engineers (England); 1969 - present.
- Smithsonian Scientific Information Exchange (SSIE); 1977 - present.
- COMPENDEX, a data base of the Engineering Index Annual, a yearly publication of engineering and engineering-related abstracts; 1970 - present.
- NSA, the Standards and Specifications data base of the National Standards Association; current standards.
- SCISEARCH, the computer file of the Science Citation Index; 1974 - present.
- DTIC (Defense Technical Information Center); reports on research and development supported by the Department of Defense; 1953-present.

In the search for data on attenuation of electromagnetic waves by building materials, typical key words used were: electrical conductivity, permittivity, building material, construction material, radiofrequency, electromagnetic shielding.

The best report for experimental data on shielding effectiveness of building materials (Brennan, et al. [4]) was obtained from the NTIS data base with the combined key words: electrical properties, electrical conductivity, permittivity, building materials, electromagnetic shielding, radiofrequency, construction. A companion paper (Garrett et al. [5]) was obtained using the same key words but omitting "construction." This selection of keywords retrieved papers on both the direct measurement of electromagnetic attenuation and on measurement of electrical properties from which attenuation could be computed. The DTIC data base search gave the Brennan [4] and Garrett [5] reports in response to the combined key words: building, construction materials, dielectric properties, electrical properties, electrical conductivity. Mrs. Schneller was adept at using different combinations of key words to make an interrogation more specific.

In our search for papers on the attenuation of electromagnetic (EM) waves by buildings (as opposed to building materials), we used key word groups such as: EM field, EM radiation, EM wave absorption, attenuation, shielding, absorption, building, structure, and construction. One of the most useful papers on building attenuation (Smith [3]) had been entered in the INSPEC data base with the key words: electromagnetic compatibility, electromagnetic fields, building, shielding, radiowave propagation. As our search progressed, we found that there was very little data on radiofrequency attenuation by conventional building materials, or even on the electrical properties of such materials.

A computer search based on key words without such qualifiers as "electromagnetic" and "radiofrequency" produces references on the attenuation of nuclear as well as electromagnetic radiation. Many papers retrieved by key words specific to "radiofrequency attenuation by buildings and building materials" concerned structures hardened to electromagnetic interference by specialized construction methods and exotic materials. Often these structures were simply shielded rooms or no more than screened enclosures. Therefore, although our search was extensive, the data we use in our building attenuation computations has come from just a few reports which are identified on the data table for each material.

2.3 Computations

The three types of materials for which we computed shielding properties are dielectrics, metal sheets, and metal meshes. We discuss these computations in that order. All expressions and computed data are in SI units.

Brennan [4] measured the real and imaginary parts of the complex permittivity of common building materials as a function of frequency from 10 Hz to 1 GHz. (Because of the variable sensitivity of the measurement apparatus as the lossiness of the materials varied, measurements often could not be made at each of the intended frequencies.) From these permittivity values, we computed the power reflection coefficient and power attenuation (dB/cm) for each material in decade steps at frequencies from 10 kHz to 10

GHz. If data was not available at a desired frequency, we plotted the given data and then interpolated or extrapolated to obtain the missing point. We now briefly discuss the relations used in our computations.

The electrical characteristics of a lossy, isotropic dielectric are given by its complex permittivity (see any text, such as Johnk [6])

$$\hat{\epsilon} = \epsilon' - j\epsilon'' \quad (2.1)$$

where the real part, ϵ' , contains information on the speed and wavelength of an EM wave in the dielectric (ϵ' corresponds to the permittivity ϵ of a lossless dielectric). The imaginary term ϵ'' accounts for the lossy nature of the dielectric and appears in expressions for the EM wave attenuation due to those losses. Electromagnetic waves are transmitted through a dielectric (lossy or not) with the velocity $v = (\mu\epsilon)^{-1/2}$, where μ is the permeability of the medium. (The free space values are denoted μ_0 , ϵ_0 , and $v = (\mu_0\epsilon_0)^{-1/2} \cong 3 \times 10^8$ m/s. Air can be considered free space.) In general, ϵ' and ϵ'' are functions of the frequency of the EM wave traversing the dielectric, as observed in Brennan's data.

Given two adjoining media of permittivities $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$, a wave in medium 1, normally incident on medium 2, has a voltage reflection coefficient ([6], p. 373)

$$\hat{\Gamma} = \frac{\hat{\eta}_2 - \hat{\eta}_1}{\hat{\eta}_2 + \hat{\eta}_1}, \quad \hat{\eta}_i = \sqrt{\frac{\hat{\mu}_i}{\hat{\epsilon}_i}} = \text{intrinsic wave impedance.}$$

Because $\mu_2 = \mu_0$ for most building materials in this report, and because medium 1 is air, we can write

$$\hat{\Gamma} = \frac{1 - \sqrt{\hat{\epsilon}_r}}{1 + \sqrt{\hat{\epsilon}_r}}, \quad \hat{\epsilon}_r = \hat{\epsilon}_2/\epsilon_0 = \text{relative permittivity,}$$

where medium 2 is the building. The power reflection coefficient is then

$$\hat{\Gamma}^2 = \left| \frac{1 - \sqrt{\hat{\epsilon}_r}}{1 + \sqrt{\hat{\epsilon}_r}} \right|^2. \quad (2.2)$$

To determine if we must retain $\hat{\epsilon}_r$ in its complex form, we write

$$\hat{\epsilon}_r = \frac{1}{\epsilon_0} (\epsilon'_2 - j\epsilon''_2) = \epsilon'_r - j\epsilon''_r$$

$$= Ae^{-j\theta}; \quad A = [(\epsilon'_r)^2 + (\epsilon''_r)^2]^{1/2}; \quad \theta = \tan^{-1}(\epsilon''_r/\epsilon'_r),$$

and so

$$(\hat{\epsilon}_r)^{1/2} = [(\epsilon'_r)^2 + (\epsilon''_r)^2]^{1/4} e^{-j\theta/2}.$$

For the materials in this report that are low-loss dielectrics ($\epsilon'' \ll 1$), we neglect ϵ''_r and approximate $\hat{\epsilon}_r$ as the real quantity ϵ'_r whose magnitude is ϵ'_r . This procedure incurs negligible error in the reflection coefficient, the worst case being 2.5% (the 30 kHz ϵ' , ϵ'' values for moist clay brick) which introduces in eq (2.2) a deviation from the true reflected power of only 0.1 dB. With the above approximation for $\hat{\epsilon}_r$, our expression for the power reflection coefficient is

$$(\Gamma)^2 = \left(\frac{1 - \sqrt{\epsilon'_r}}{1 + \sqrt{\epsilon'_r}} \right)^2. \quad (2.3)$$

As an EM wave traverses a lossy dielectric, it is attenuated; that is, the wave amplitude decreases with distance. We represent this wave as

$$E_x(z) = E_0 e^{-\gamma z} = E_0 e^{-(\alpha + j\beta)z} = E_0 e^{-\alpha z} e^{-j\beta z}$$

where the E-field vector is in the x direction, and the wave propagates along the positive z axis. The exponential factor, $E_0 e^{-\alpha z}$, represents the wave-amplitude attenuation with distance; α is the attenuation constant. The ratio of amplitudes at points z and z + ℓ is

$$\frac{E_0 e^{-\alpha z}}{E_0 e^{-\alpha(z + \ell)}} = e^{\alpha \ell}.$$

The attenuation of power density in the wave over the distance ℓ is the square of this ratio and is $e^{2\alpha \ell}$. In decibels, this attenuation is

$$\text{attenuation (dB)} = 10 \log e^{2\alpha \ell} = 10 \times 0.4342 \ln e^{2\alpha \ell} = 8.684\alpha \ell$$

and so

$$\text{attenuation (dB/length)} = 8.684\alpha. \quad (2.4)$$

For a wave of angular frequency ω in a dielectric of complex permittivity $\epsilon' - j\epsilon''$ and permeability μ ($= \mu_0$ for dielectrics considered here) we have (Johnk [6], p. 173)

$$\alpha = \frac{\omega \sqrt{\mu \epsilon'}}{\sqrt{2}} [\sqrt{1 + (\epsilon''/\epsilon')^2} - 1]^{1/2}. \quad (2.5)$$

Brennan has given his values of ϵ' , ϵ'' as ϵ'/ϵ_0 ($= \epsilon'_r$) and ϵ''/ϵ_0 ($= \epsilon''_r$) which he calls dielectric constant ("relative permittivity," in Johnk) and dissipation factor (Johnk's dissipation factor is ϵ''/ϵ'). Writing α in terms of these quantities (where $c \sim 3 \times 10^8$ m/s)

$$\alpha = \frac{\omega \sqrt{\mu_0 \epsilon_r' \epsilon_0}}{\sqrt{2}} \left[\sqrt{1 + \left(\frac{\epsilon_r''}{\epsilon_r'} \right)^2} - 1 \right]^{1/2} = \frac{2\pi f \sqrt{\epsilon_r'}}{\sqrt{2} c} \left[\sqrt{1 + \left(\frac{\epsilon_r''}{\epsilon_r'} \right)^2} - 1 \right]^{1/2}$$

$$= 1.48 \times 10^{-8} f \sqrt{\epsilon_r'} \left[\sqrt{1 + \left(\frac{\epsilon_r''}{\epsilon_r'} \right)^2} - 1 \right]^{1/2} . \quad (2.6)$$

In eq (2.5), the factor $\omega \sqrt{\mu \epsilon} = 2\pi/\lambda$; thus, the dimension of α is $(\text{length})^{-1}$. In metric (S.I.) units, α has the dimension $(\text{meter})^{-1}$, and eq (2.4) will be in dB/m. For our purposes, a more reasonable dimension is dB/cm, and so our values for attenuation were computed from the expression

$$\text{attenuation (dB/cm)} = 0.08684\alpha , \quad (2.7)$$

with α obtained from eq (2.6).

For metals and materials with significant conductivity, σ , the complex permittivity is more appropriately written

$$\hat{\epsilon} = \epsilon - j \frac{\sigma}{\omega} . \quad (2.8)$$

Following the discussion in Johnk ([6], ch. 3), the corresponding form for α is

$$\alpha = \frac{\omega \sqrt{\mu \epsilon}}{\sqrt{2}} \left[\sqrt{1 + \left(\frac{\sigma}{\omega \epsilon} \right)^2} - 1 \right]^{1/2} . \quad (2.9)$$

For metals $\sigma \sim 10^7$ mhos/m. Therefore, even at 10 GHz, $\left(\frac{\sigma}{\omega \epsilon} \right)^2 \gg 1$, and α can be written

$$\alpha = \frac{\omega \sqrt{\mu \epsilon}}{\sqrt{2}} \sqrt{\frac{\sigma}{\omega \epsilon}} = \sqrt{\frac{\omega \mu \sigma}{2}} = \sqrt{\frac{\omega \mu_r \mu_0 \sigma_r \sigma_c}{2}}$$

where μ_r is relative permeability, σ_r is conductivity relative to copper, and σ_c is the conductivity of copper. Substituting $\mu_0 = 4\pi \times 10^{-7}$ farads/meter and $\sigma_c = 5.80 \times 10^7$ mhos/meter, we have

$$\alpha = 15.13 \sqrt{f \mu_r \sigma_r}$$

and so

$$\text{attenuation (dB/dm)} = 0.08684\alpha = 1.314 \sqrt{f \mu_r \sigma_r} . \quad (2.10)$$

This is the expression in Denny ([7], p. 5-6) for the attenuation of an EM wave traversing a metal sheet 1 cm thick. With eq (2.10) and values for μ_r and σ_r from Table 5-2 in Denny, we computed attenuation values for iron, copper, and aluminum sheets.

For a plane wave normally incident on a surface, the incident, reflected, and transmitted powers are related as

$$P_i = P_r + P_t ,$$

where P_t is the wave power density (w/cm^2) just across the interface and before the wave traverses any of the medium into which it has just passed. Then

$$1 = \frac{P_r}{P_i} + \frac{P_t}{P_i} = (\Gamma_r)^2 + (\Gamma_t)^2$$

and so

$$(\Gamma_r)^2 = 1 - (\Gamma_t)^2 \quad (2.11)$$

where Γ_r and Γ_t are the magnitudes of voltage reflection and transmission coefficients, and their squares are power reflection and transmission coefficients. In Denny ([7], p. 5-5), the reflection loss in dB is given as

$$R = -10 \log (\Gamma_t)^2 . \quad (2.12)$$

For iron and copper, R values are tabulated in Campi ([8], p. 28), and presented graphically for iron, copper, and aluminum in Denny (p. 5-15). Solving eq (2.12) for $(\Gamma_t)^2$, we have

$$(\Gamma_t)^2 = 10^{-R/10}$$

and we write the power reflection coefficient as

$$(\Gamma_r)^2 = 1 - 10^{-R/10} . \quad (2.13)$$

In Denny [7] and Campi [8], the smallest value of R is 57 dB from 10 kHz to 10 GHz. Therefore, we set the power reflection coefficient to unity over this frequency range for iron, copper and aluminum.

The attenuation of plane EM waves incident normally on metal wire meshes has been computed by Jakubec and Ohta [9], and we give their attenuation values in Tables 2.9 and 2.10 for galvanized steel and copper meshes. Jarva [10] has done some plane wave attenuation computations from the same equations and his values are close to those of Jakubec and Ohta. Some measured attenuations quoted by Jarva support the computed plane wave values. The theoretical expressions employed in both reports are identical, though Jarva gives their derivations.

The mesh attenuation ("insertion loss" in Jarva) computed by Jakubec and Ohta is the same quantity as R in eq (2.12). Using eq (2.13) and attenuation values in Tables 2.9 and 2.10 for R, we see that we are justified in setting the mesh reflection coefficients to unity.

The attenuation and reflection coefficients of a reinforced concrete wall are almost totally due to the reinforcing bars ("rebars") within the concrete. The low shielding effectiveness of concrete alone is seen in our computations of attenuation and reflection coefficients from Brennan's permittivity data for moist mortar (Table 2.2). ("Moist" means the mortar samples were exposed to a saturated atmosphere for one day prior to measurement. Some samples were measured dry: they were baked at 140°F for about 20 hours prior to measurement. However, the moist samples had electrical properties most similar to those of materials in field conditions.) There are many types of concrete reinforcing structures. In some cases, parallel bars without cross members are used; the bars may be horizontal or vertical. When a mesh is

used, the vertical and horizontal bars may have the same spacing and the same bar diameters, or the vertical bars may be heavier and closer together. (Rebar diameters range from 0.95 cm to 6.4 cm.) Thicker walls may have two reinforcing layers: a 20-cm thick concrete wall could have a rebar layer 5 cm in from each face. Thus, knowing only that a concrete wall is reinforced, one cannot be sure of the reinforcing configuration. We use a square mesh to illustrate how rebar shielding may vary with frequency.

The equations presented in references 7, 9, and 10 for the attenuation of plane EM waves by metal screens are based on the transmission of evanescent modes through a waveguide below cutoff. However, reinforcing meshes in concrete have such large openings that cutoff occurs at much lower frequencies, well into the frequency range (10 kHz - 10 GHz) considered in this report. Therefore, we compute the power reflection coefficient for rebar meshes using an equation developed by MacFarlane ([11], p. 1527) for the voltage reflection coefficient for plane waves incident on an infinite parallel-wire grid, the plane of polarization parallel to the grid. Hill and Wait [12] obtain the same expression in their analysis of the scattering of a transient plane wave by a periodic grating. In both these treatments the scattering structure is an infinite set of equally-spaced parallel wires; there are no cross members, unlike the rebar mesh we are considering. However, Hill and Wait [13, 14] show that, for normal incidence, the reflection coefficient obtained by MacFarlane [11] and by Hill and Wait [12] is applicable to scattering from a mesh. For waves at normal incidence, the two crossed grids forming the mesh decouple and interact with the waves as separate, independent parallel-wire grids, each responding only to the E-field component along it. Thus, when the plane of polarization is aligned with one grid, the other has no interaction with the waves and drops out of the analysis. For normal incidence, the mesh field equations obtained by Hill and Wait [13, 14] yield the MacFarlane expression for the reflection coefficient of a parallel-wire grid.

The decoupling of crossed grids of parallel wires is also shown by Kontorovich [15] and Astrakhan [16], but only for the long-wavelength condition ($d \ll \lambda$, d = separation distance between wires of a grid). (Their results cannot be used for wavelengths equal to or less than d , as in our rebar computations.) The analysis contains a term proportional to the electrical resistance between the grids at the points where the wires of one grid are bonded to those of the other. The reflection coefficient for waves at normal incidence does not depend on this term so the grids are, in effect, independent elements of the mesh. For $d \ll \lambda$, MacFarlane's expression for the normal incidence reflection coefficient of a parallel-wire grid reduces to the expression for a mesh [15, 16].

For wavelengths longer than the wire spacing, the power reflection coefficient obtained from MacFarlane for 0° angle-of-incidence is

$$|\hat{\Gamma}_r|^2 = \frac{1}{1 + \left(\frac{2d}{\lambda}\right)^2 [F(\frac{d}{\lambda}, 0^0) + \ln \frac{d}{2\pi a}]^2} \quad (2.14)$$

where d = wire spacing, a = wire radius. For $\frac{d}{\lambda} < 1$ (d = wire spacing), $F(\frac{d}{\lambda}, 0^0)$ is a real factor given graphically in references 11 and 12. For short wavelengths ($\frac{d}{\lambda} > 1$), F becomes complex ($\hat{F} = F_r + j F_i$), and values for F_r and F_i are also plotted [12]. The power reflection coefficient is then

$$|\hat{\Gamma}_r|^2 = \frac{1}{(1 - \frac{2d}{\lambda} F_i)^2 + \left(\frac{2d}{\lambda}\right)^2 (F_r + \ln \frac{d}{2\pi a})^2} \quad (2.15)$$

We have used eqs (2.14) and (2.15) to compute the plane-wave power reflection coefficient for a square rebar mesh.

Campi [8] discussed the shielding effectiveness of a rebar mesh having $d = 35.6$ cm and $a = 2.2$ cm. We used these dimensions in computing rebar power reflection coefficients from eqs (2.14) and (2.15). The definition of shielding effectiveness (SE) for any shielding material is ([7], p. 5-2)

$$SE = 10 \log \frac{P_1}{P_2} = -10 \log \frac{P_2}{P_1} = -10 \log (\Gamma_t)^2 = \text{attenuation (dB)}$$

where (P_1, P_2) = power density (without, with) the shield in place. Then we use eq (2.11) and compute the rebar attenuation from the expression

$$\text{attenuation (dB)} = -10 \log (\Gamma_t)^2 = -10 \log (1 - (\Gamma_r)^2). \quad (2.16)$$

We summarize our computation equations:

- Power reflection coefficient for a dielectric sheet: eq (2.3).
- Power attenuation within a dielectric sheet: eqs (2.6), (2.7).
- Power reflection coefficient for a metal sheet: eq (2.13).
- Power attenuation within a metal sheet: eq (2.10).
- Power reflection coefficient for a metal screen: eq (2.13), using eq (2.12).
- Power reflection coefficient for a rebar mesh: eqs (2.14), (2.15).
- Power attenuation (insertion loss) for a rebar mesh: eq (2.16).

2.4. Uncertainty

For the quantities we have computed (reflection coefficient, attenuation per unit length), we used expressions derived for plane waves at normal incidence. Into these expressions we put measured values for the real and imaginary parts of the complex permittivity. Thus, our computed quantities have uncertainties originating in the measured data we obtained in our literature search. (The reports from which this data was obtained do not give measurement uncertainties.)

However, we must include additional uncertainties in our computed data to account for other factors which influence reflection coefficients and attenuations. For example, although we have assumed normal incidence, waves may be incident at angles from 0° to 90° . A concern? Yes. Fresnel's reflection equations tell us that the reflection coefficient depends on the angle of incidence and the orientation of the plane of polarization with respect to the plane of incidence. The power in a reflected wave may also vary with surface dampness. Another factor adding to the uncertainty in our data is the dependence of wood permittivity on temperature and relative humidity [17], on the angle between the plane of incidence and the wood grain [17], and on chemicals used in treating the wood [18]. There may be other less determinate factors contributing to deviations from our computed data, factors such as manufacturing differences, age of materials, and surface weathering.

Because our computation equations were derived from plane-wave models, they contribute further to the uncertainty of our computed data. As discussed in Denny ([7], section 5.3.2), the reflection coefficient depends on the intrinsic impedance of the incident wave. Plane waves have an intrinsic impedance of about 377Ω , while waves in the near fields of loop and dipole antennas are not planar and have lower and higher impedances, respectively. These three types of electromagnetic field are all different in their reflection loss versus frequency curves. The reflection loss of high and low impedance fields also depends on the distance of the reflecting surface from the source antenna. Thus, the reflection coefficients we have computed will differ from the true reflection coefficient when the incident wave is something other than plane.

The program MASTER computes only worst-case values for building attenuation and does not do an error analysis. The latter would be of little use considering our lack of detailed information on radiation environments and building structure and contents. However, to acknowledge the "unknowable" uncertainties introduced into our data by the various factors we have discussed, we suggest the following broad uncertainty estimates for the quantities specified:

- 1% - the essentially infinite attenuations and unity reflection coefficients of metal sheets and meshes. This small uncertainty indicates that these materials have a nearly constant effect versus frequency and changing environment.
- 10% - the attenuations of the dielectric materials (e.g., glass, brick, wood). We have assigned this higher uncertainty because these materials, nearly transparent up to microwave frequencies, are the major reason why fields so easily penetrate conventional buildings. Variations in the electrical properties of these materials will alter (though only slightly) the power density of waves passing through them into building interiors.
- 100% - the reflection coefficients of the dielectric materials. This large uncertainty should include most variations in reflection coefficient with angle of incidence, material properties, and environmental conditions.

Note that these uncertainties are in the reflected and attenuated powers and not in the decibel per centimeter values for attenuation.

When we compute the plane-wave shielding effectiveness of a layer of construction material in a wall, the reflection coefficient of the material tells us how much incident power is turned back by the layer, and the attenuation (dB/cm) tells us how much power is absorbed within the material of the layer. (By "power," we mean power density in the wave, e.g., W/cm².) Reflection occurs not only at the front, but also at the back surface of the layer. Depending on the reflection coefficient and the attenuation, there may be enough power in the back surface reflection that the succeeding multiple internal reflections within the layer must be considered in determining the net transmission through the layer and the net reflection from the layer. (In this regard, solid metal shields can be neglected. They have such high attenuation that little or no electromagnetic field reaches the back surface of the shield.) Common building materials are dielectrics with very low attenuation and reflection coefficients, and much of the power in the incident wave passes through such a material. Therefore, we must decide if it is sufficient to consider only the front surface reflection, or if the multiple internal reflections within a dielectric layer should be taken into account.

At any instant, the internal reflections within a dielectric sheet produce an infinite series of waves leaving the front and back surfaces of the sheet. The vector addition of the fields in these waves gives the net reflected and transmitted wave. However, to simplify the treatment and still get an estimate of the reflected and transmitted power, we assume constructive interference between all the emerging waves and so add their powers to obtain the total reflected and transmitted power. For low-loss dielectrics, we neglect attenuation within the material. This procedure gives the total reflected power

$$P_r = \frac{2(\Gamma_r)^2}{1 + (\Gamma_r)^2} P_o.$$

where P_o is the power density of the incident wave, and $(\Gamma_r)^2$ is the power reflection coefficient. The total transmitted power is

$$P_t = \frac{1 - (\Gamma_r)^2}{1 + (\Gamma_r)^2} P_o.$$

For our dielectric materials, $(\Gamma_r)^2 \ll 1$, and we can write

$$P_r \cong 2 (\Gamma_r)^2 [1 - (\Gamma_r)^2] P_o$$

and

$$P_t \cong [1 - (\Gamma_r)^2]^2 P_o.$$

To first order in $(\Gamma_r)^2$

$$\begin{aligned} P_r &\cong 2(\Gamma_r)^2 P_o \\ P_t &\cong [1 - 2(\Gamma_r)^2] P_o. \end{aligned} \quad (2.17)$$

For a single surface (i.e., ignoring multiple reflections caused by a second surface), the reflected and transmitted powers are obtained by inspection

$$\begin{aligned} P_r &= (\Gamma_r)^2 P_o \\ P_t &= [1 - (\Gamma_r)^2] P_o. \end{aligned} \quad (2.18)$$

Even though the pairs of eqs (2.17) and (2.18) differ slightly in form, $(\Gamma_r)^2$ is so small that the effect of a second surface on P_r is generally negligible; and, in both cases, P_t is so slightly different from P_o that, again, we ignore the effect of a second surface. Therefore, we consider only a single reflecting/transmitting surface for each material in a building wall. Any error incurred by this assumption will be covered by the 100% uncertainty we have assigned to the reflection coefficients for dielectric construction materials.

2.5 Reflection Coefficient and Attenuation Data for Selected Building Materials

The data in the following fourteen tables has been entered into the data file MATTER and is ready for use in the building attenuation computations performed by the program MASTER. We preface the tables with these comments:

- The null material (M01) must be used as the "material" of an open doorway or an open, unscreened window. As a formality whenever material M01 is needed, the user must enter into the data file HOLES a material thickness T of 1 cm.
- Dry wall (wall board, sheet rock) is mainly plaster of Paris (material M03).
- The word "moist" (materials M02, M03, M06, M07) does not imply "soft", "fresh", or "uncured", but only that material samples measured after 24 hours in a saturated atmosphere (as opposed to samples baked dry) had electrical properties more similar to the same materials in field conditions.
- "Clay brick" (material M06) refers to the brick commonly used in homes and buildings.
- All common lumber and plywood have a very low reflection coefficient and attenuation (dB/cm). Therefore, the material data tables contain only Douglas fir and fir plywood as representative types to be used for any wood or plywood the user may encounter as building materials.
- Because the attenuation by a metal screen (materials M09, M10, M11) is actually an insertion loss given in dB instead of dB/cm, the mesh thickness is not required. However, as a formality to satisfy the computation program MASTER, the user must enter into the data file BxxxxxT a mesh thickness T of 1 cm.
- The assigned uncertainties are in the transmitted and reflected powers, and must not be applied to the attenuation in dB/cm.

Table 2.1: Null Material (Mat'l. No. M01) (Thickness T = 1 cm)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	0	0
10^5	0	0
10^6	0	0
10^7	0	0
10^8	0	0
10^9	0	0
10^{10}	0	0

Material: Fictitious; a formality, required by the program MASTER, used to represent the "material" of passageways, open doorways, and open, unscreened windows.

Table 2.2: Moist Mortar (Mat'l. No. M02)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	1.22×10^{-6}	0.23
10^5	7.54×10^{-6}	0.17
10^6	3.55×10^{-6}	0.16
10^7	7.70×10^{-5}	0.13
10^8	1.89×10^{-3}	0.10
10^9	1.12×10^{-2}	0.055
10^{10}	0.13*	0.03*

Material: Moist mortar; 6.5 gal. H₂O/94 lb. sack of cement; Portland cement-aggregate ratio: 1/3.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.3: Plaster of Paris (Mat'l. No. M03)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	2.98×10^{-7}	0.063
10^5	4.41×10^{-7}	0.059
10^6	1.50×10^{-6}	0.084
10^7	2.58×10^{-6}	0.076
10^8	4.84×10^{-5}	0.063
10^9	7.6×10^{-4}	0.007
10^{10}	7.6×10^{-3} *	0.007*

Material: Moist plaster of Paris (main component of dry wall).

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.4: Douglas Fir (Mat'l. No. M04)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	4.28×10^{-7}	0.047
10^5	2.59×10^{-6}	0.041
10^6	4.35×10^{-6}	0.063
10^7	1.10×10^{-4}	0.050
10^8	1.98×10^{-3}	0.025
10^9	2.0×10^{-2}	0.019
10^{10}	0.22*	0.014*

Material: Douglas fir.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.5: Fir Plywood (Mat'l. No. M05)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	1.15×10^{-6}	0.068
10^5	6.77×10^{-6}	0.048
10^6	8.31×10^{-6}	0.074
10^7	1.24×10^{-4}	0.036
10^8	2.15×10^{-3}	0.014
10^9	2.6×10^{-2}	0.013
10^{10}	0.30*	0.010*

Material: Fir plywood.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.6: Clay Brick (Mat'l. No. M06)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	2.20×10^{-6}	0.13
10^5	1.02×10^{-5}	0.072
10^6	$1.4 \times 10^{-4*}$	0.051*
10^7	$2.5 \times 10^{-3*}$	0.029*
10^8	5.72×10^{-3}	0.014
10^9	5.72×10^{-3}	0.014
10^{10}	$5.7 \times 10^{-3*}$	0.014*

Material: Moist clay brick.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Interpolated or extrapolated.

Table 2.7: Cinder Block (Mat'l. No. M07)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	1.74×10^{-6}	0.17
10^5	8.30×10^{-6}	0.14
10^6	3.85×10^{-5}	0.13
10^7	2.82×10^{-4}	0.055
10^8	5.71×10^{-3}	0.013
10^9	5.71×10^{-2}	0.013
10^{10}	0.57*	0.013*

Material: Moist cinder block (Featherlite).
 Data Source: Brennan [4].
 Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.8: Glass (Mat'l. No. M08)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	1.36×10^{-8}	0.20
10^5	2.28×10^{-7}	0.18
10^6	2.88×10^{-6}	0.19
10^7	3.95×10^{-5}	0.19
10^8	4.00×10^{-4}	0.15
10^9	5.04×10^{-3}	0.082
10^{10}	7.1×10^{-2} *	0.015*

Material: Glass (type not specified). We assume "window glass" because the Brennan report concerns only building materials.

Data Source: Brennan [4].
 Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.9: Steel Mesh (Mat'l. No. M09) (Thickness T = 1 cm)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	141.0*	1.0
10^5	132.0*	1.0
10^6	114.0	1.0
10^7	94.1	1.0
10^8	44.0*	1.0
10^9	54.1	1.0
10^{10}	34.0*	1.0

Material: Galvanized steel mesh (24 × 24).
 Data Source: Jakubec and Ohta [9].
 Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

*Interpolated or extrapolated.

Table 2.10: Copper Mesh (Mat'l. No. M10) (Thickness T = 1 cm)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	106.0*	1.0
10^5	110.0*	1.0
10^6	104.2	1.0
10^7	88.0	1.0
10^8	68.0*	1.0
10^9	48.4	1.0
10^{10}	28.0*	1.0

Material: Copper mesh (20 × 20).

Data Source: Jakubec and Ohta [9].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

*Interpolated or extrapolated.

Table 2.11: Reinforcing Bar Mesh (Mat'l. No. M11) (Thickness T = 1 cm)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	1000.0*	1.0
10^5	1000.0*	1.0
10^6	1000.0*	1.0
10^7	1000.0*	1.0
10^8	12.2	0.94
10^9	0.22	0.05
10^{10}	0	0

Material: Reinforcing bar square mesh; 35.6 cm on centers; bar diameter = 4.3 cm.

Data Source: Hill and Wait [12]; MacFarlane [11].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

*To represent infinite attenuation as computed from unity power reflection coefficient.

Table 2.12: Iron Sheet (Mat'l. No. M12)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	$1.71 \times 10^{+3}$	1.0
10^5	$5.42 \times 10^{+3}$	1.0
10^6	$1.43 \times 10^{+4}$	1.0
10^7	$3.83 \times 10^{+4}$	1.0
10^8	$5.42 \times 10^{+4}$	1.0
10^9	$1.21 \times 10^{+5}$	1.0
10^{10}	$5.42 \times 10^{+4}$	1.0

Material: Iron sheet.

Data Source: Denny [7]; Campi [8].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

Table 2.13: Aluminum Sheet (Mat'l. No. M13)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	$1.03 \times 10^{+2}$	1.0
10^5	$3.24 \times 10^{+2}$	1.0
10^6	$1.03 \times 10^{+3}$	1.0
10^7	$3.24 \times 10^{+3}$	1.0
10^8	$1.03 \times 10^{+4}$	1.0
10^9	$3.24 \times 10^{+4}$	1.0
10^{10}	$1.03 \times 10^{+5}$	1.0

Material: Aluminum sheet.

Data Source: Denny [7]; Campi [8].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

Table 2.14: Copper Sheet (Mat'l. No. M14)

<u>Frequency (Hz)</u>	<u>Attenuation (dB/cm)</u>	<u>Power reflection coefficient</u>
10^4	$1.31 \times 10^{+2}$	1.0
10^5	$4.16 \times 10^{+2}$	1.0
10^6	$1.31 \times 10^{+3}$	1.0
10^7	$4.16 \times 10^{+3}$	1.0
10^8	$1.31 \times 10^{+4}$	1.0
10^9	$4.16 \times 10^{+4}$	1.0
10^{10}	$1.31 \times 10^{+5}$	1.0

Material: Copper sheet.

Data Source: Denny [7]; Campi [8].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

3. CALCULATION FOR BUILDING SHIELDING EFFECTIVENESS -- THEORY

3.1 Introduction

This chapter discusses the theoretical basis for the calculation of building shielding effectiveness. After this introduction, there are four sections which consider a definition of the problem, the input data that will be used in the calculation, the actual mathematical approach used to calculate building shielding effectiveness, and window and door resonances.

3.2 Definition of Problem

The purpose of this project is to calculate how well a building shields from external electromagnetic interference. Automated on a computer, this calculation is to provide a worst case estimate of the electromagnetic power level appearing in a room of a building given the physical construction and dimensions of the building and given a known external incident field strength. The estimate must be useful over a wide frequency range (10 kHz - 10 GHz) and must be general enough so that it can be used effectively on several hundred differently constructed buildings.

3.3 Input Data Available

In any given calculation of electromagnetic shielding effectiveness, it is always desirable to have sufficient, high quality data to make the calculation yield a precision result. For example, the best data possible would be to actually measure a given building for electromagnetic shielding effectiveness using external antennas to launch a known field and calibrated antennas to measure the field inside of the building. The calculation would then be trivial -- the ratio of the launched field versus the measured internal field would be calculated directly. In fact such measurements have been performed and have yielded excellent results. Such field measurements made during this project are detailed in Chapter 6 of this report.

Direct measurements of electromagnetic shielding effectiveness, although highly effective in characterizing a building, have the disadvantage of being time consuming and expensive. A typical field measurement of one building at one site can take three or four engineers from three to five days to complete. When three to five hundred buildings are contemplated, it clearly becomes difficult to obtain actual field data in an efficient manner. Another approach is to take "less expensive" data that can be used in a model that will give an estimate of the shielding effectiveness. This estimate may not be as accurate as actual field data, but it may give sufficient accuracy to make a cost effective analysis of a building. This section describes the data that will be used in the estimation of building measurements.

For a particular calculation for a given building, the input data is restricted to the actual physical construction details of the building. The input data include: 1) building construction materials; 2) the dimensions of the building; and, 3) the construction practices involved in erecting the building. These three categories of data are to be obtained either from building drawings which were used in the original construction and which have been properly updated with modifications, or by actual physical measurements on the building itself. Although it will be less accurate than direct EMI measurements, the estimate of EM shielding effectiveness based on the physical construction of the building should be sufficiently accurate to determine the suitability of most buildings for electromagnetic shielding. In the few cases where the calculated estimate is perhaps ambiguous, actual EMI measurements could be performed. A test plan for field measurements of EMI for these marginal cases is presented in the appendix. Details and methods for gathering the actual physical data and putting it into a form usable in the present computer model will be presented in Chapter 5, "User's guide to the computer programs."

3.4 Approach

To estimate the electromagnetic shielding ability of a building, the following approach is taken. The energy entering the building from external sources is calculated for each room by considering the direct penetration through all the walls, windows and doors. The energy in each room can then go to the other rooms through internal walls and doors, as well as going back outside. As the energy continues to flow through each room, the power in each room reaches a steady state condition corresponding to a balance between the energy flowing in and out. To calculate the steady state power level, it is necessary to first consider a model that will describe the flow of the electromagnetic energy through the building. The assumptions in this model listed in figure 3.1, and graphically shown in figure 3.2, must also be considered.

The first assumption is that the input electromagnetic radiation will consist of plane waves with normal incidence on a particular external wall. The calculation will be repeated five times corresponding to the four horizontal incidence directions (i.e., North, South, East and West) and the one vertical direction (i.e., from above). The sixth direction, from below or through the ground will be ignored.

The first assumption is used to calculate the power entering a room through an external wall from the outside. If the energy density of the incident field is represented by P with units of joules/m^3 , the energy, E_{IN} , entering a room in one second will be

$$E_{IN} = tPcA \quad (3.1)$$

where t is the transmission coefficient/unit length of the external wall and c is the speed of the electromagnetic radiation (the speed of light). The term A represents the area of the external wall. If the room has a door in the wall, eq (3.1) becomes

$$E_1 = t_1 P c A_1 + t_2 P c A_2 \quad (3.2)$$

where t_2 and A_2 represent the transmission coefficient and area of the door.

The second assumption is that there will be no refraction by the building of the incident plane wave, and the incoming energy will only enter those walls which are normal to the incoming radiation. This effect can also be considered a shadowing effect.

The third assumption deals with multiple layers on a wall. It states that the composite transmission coefficient of a multi-layered wall will be represented as the product of the individual transmission coefficients of the layers or

$$t_{\text{composite}} = \prod t_i \quad (3.3)$$

where the t_i 's represent the transmission coefficient of the layers. This is really a simplification that ignores internal reflections in the walls and the related interference effects.

The fourth assumption considers how the energy in a room scatters and flows to the other rooms. This assumption states that the energy in a room will be isotropically scattered in all directions. Hence, in a cubic room where the four walls, the floors and ceilings all have the same area, one sixth of the room's energy would hit each of the six surfaces equally.

To find the transmission of energy from one room to another, consider figure 3.3, a building with two rooms. The first room has an energy density N , (J/m^3). From assumption four, this energy is isotropically scattered in all directions equally. The fraction, F , of the energy scattered against the internal wall will be

$$F = \frac{A_w}{A_{1,\text{total}}} \quad , \quad (3.4)$$

where A_w is the internal wall area, and $A_{1,\text{total}}$ is the total area of the walls, floor, and ceiling in room 1. The energy that is transmitted per second from room 1 to room 2 will be

$$E_{1,2} = t N_1 c F A_w \quad (3.5)$$

where the subscripts on E represent flow from room 1 to room 2. Substituting for F yields

$$E_{1,2} = t N_1 c \frac{A_w^2}{A_{1,t}} \quad (3.6)$$

Note that the energy flowing from room 2 to room 1 will be

$$E_{2,1} = t N_2 c \frac{A_w^2}{A_{2,t}} \quad (3.7)$$

which differs from eq (3.6) because the room 2 energy density, N_2 , is different and the total area of room 2, $A_{2,t}$, is different. Using eqs (3.1) and (3.6) the energy transmitted between the outside to the

rooms, and from room to room, can be described. We must now consider absorption and reflection for the walls, which brings us to the fifth assumption.

The fifth assumption states that wall reflections will be ignored unless the frequency of the radiation is near a resonance of the room. Outside of resonance, any radiation hitting a wall that is not transmitted will be absorbed or

$$D = 1 - t \quad (3.8)$$

where D is the absorption coefficient and t is the transmission coefficient. When the frequency is near a room resonance then reflections will be accounted for and

$$D = 1 - t - r \quad (3.9)$$

where r now represents the reflection coefficient.

The frequency range of a room resonance will be between F_{low} and F_{high} where

$$F_{low} = \frac{c}{2} \left[\frac{1}{B^2} + \frac{1}{C^2} \right]^{\frac{1}{2}}$$

and

$$F_{high} = \frac{c}{2} \left[\left(\frac{3}{A} \right)^2 + \left(\frac{3}{B} \right)^2 + \left(\frac{3}{C} \right)^2 \right]^{\frac{1}{2}}$$

where A, B, and C represent the three dimensions of the room with B and C being the longest. Experience indicates that modes above 3, 3, 3 do not contribute significantly to resonance fields in a room. Therefore, these higher order modes are ignored in this model.

Note that in resonance, eq (3.9) reduces the absorption coefficient and hence reduces the effective energy loss from the room. This is equivalent to reflecting it back into the room.

The energy losses in a room can now be found by replacing t in eq (3.7) with D from either eq (3.8) or (3.9) and obtaining

$$L_{1,2} = D N_1 c \frac{A_w^2}{A_{1,t}} \quad (3.10)$$

where L represents the energy lost from the room.

The sixth assumption states that in a calculation of the steady state energy distribution within a structure, the E or H field is computed by assuming free space impedance. To calculate the steady state energy in a room, we first write down an equation which represents the change in energy with respect to time or

$$\frac{dN_1 V_1}{dt} = G - L \quad (3.11)$$

where V_1 is the volume of the room and G represents the sum of all the energy gains into the room as represented by either eqs (3.2), (3.6) or (3.7). Note that a room can gain energy from the outside (eq (3.2)) and from other rooms (eqs (3.6) or (3.7)). In eq (3.11), L represents the energy lost from a room and is taken from eq (3.6) for energy transmitted out or eq (3.10) for energy absorbed in a wall. Note that for M rooms, there will be M eq (3.11)'s.

For the steady state condition eq (3.11) will equal zero. Hence there will be M eq (3.11)'s equal to zero. Since the M eq (3.11)'s have only M unknowns (the M energy levels in each room) there will be a unique solution which gives the M energy levels in the rooms.

3.5 Input Resonances

Room resonances have already been considered by adding the reflection coefficients of a wall in eq (3.9). An additional input resonance is also considered for windows or doors in the following manner. If a window or door has a metal frame, then the transmission coefficient for the window is increased by 20 dB if the incoming frequency is near a window resonance. The frequency range for a window resonance will be defined as lying between F_{low} and F_{high} where

$$F_{\text{low}} = \frac{c}{2B}$$

$$F_{\text{high}} = \frac{c}{2} \left[\left(\frac{3}{B} \right)^2 + \left(\frac{3}{A} \right)^2 \right]^{\frac{1}{2}}$$

where A and B are the smallest and largest dimensions of the windows. Note that resonances above mode 3,3 are ignored.

1. Plane wave incident on surface(s) of structure. Use attenuation of wall material(s) to calculate transmission.
2. Ignore external refraction by building.
3. Disregard multiple internal reflections within wall(s).
4. Energy reaching inside of structure is scattered isotropically. Energy leaves structure through all available surfaces.
5. Reflections within room from walls, ceiling, etc., are only considered if frequency is near room resonance(s). Only first three resonant modes are considered. Input coupling resonances are also considered for metal door and window frames by increasing transmission by 20 dB near resonant frequency.
6. Steady state energy distribution within structure allows E or H field to be computed (free space impedance assumed).

FIGURE 3.1 ASSUMPTIONS USED IN MODEL.

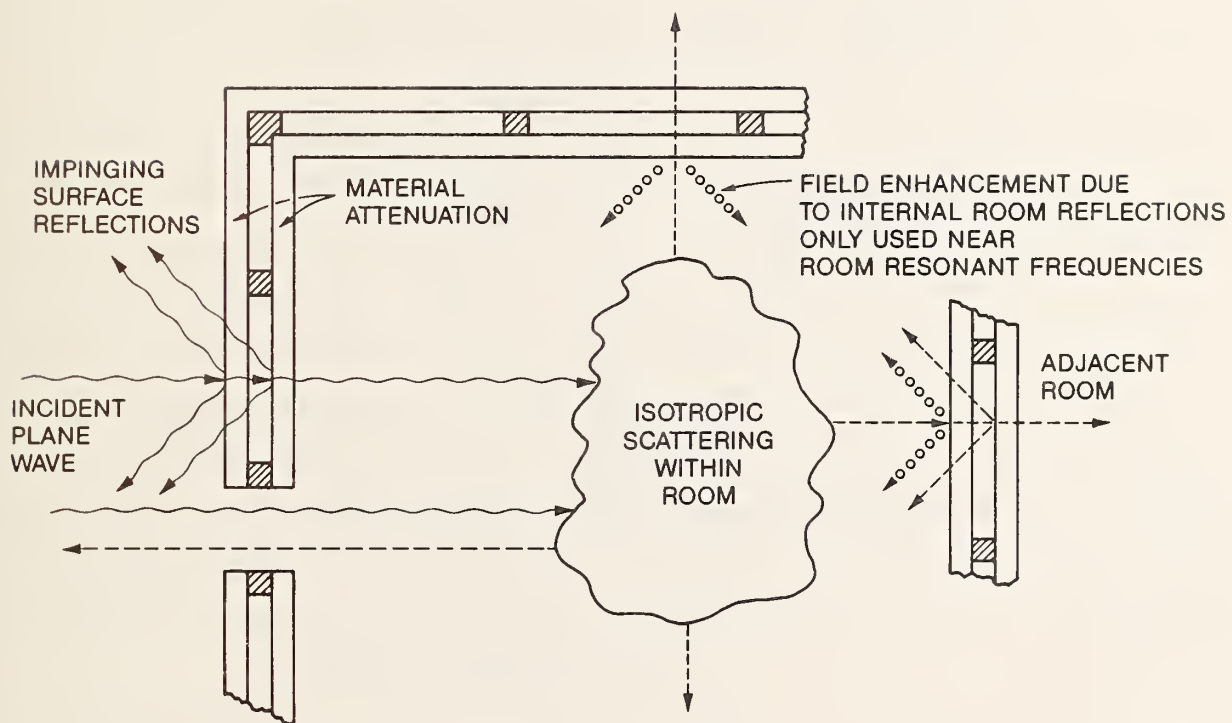


FIGURE 3.2 ENERGY FLOW ANALYSIS.

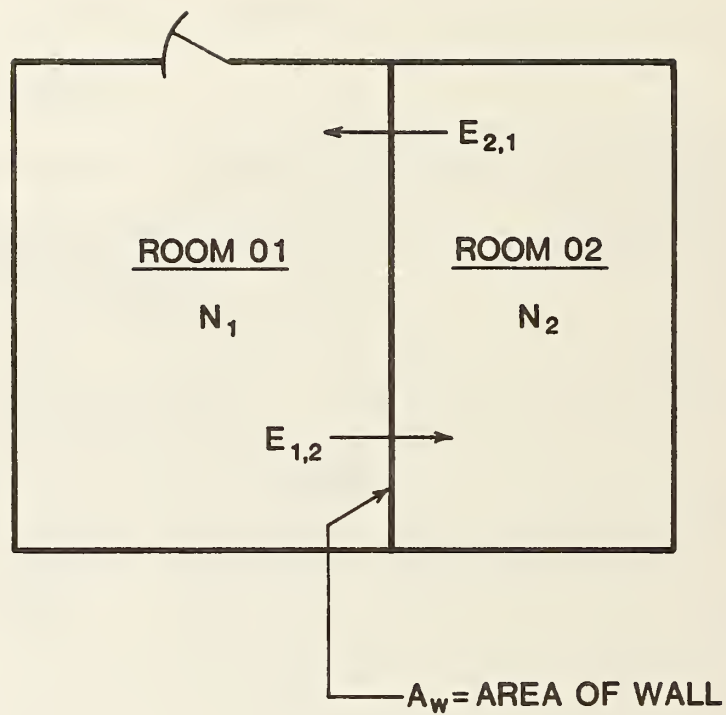


FIGURE 3.3 ENERGY FLOW BETWEEN TWO ROOMS.

4. COMPUTER PROGRAM DESCRIPTIONS

4.1 Introduction

This chapter provides a description of the computer programs used in the calculation of electromagnetic shielding effectiveness of buildings. The presentation is aimed at the programmer level. The intention is to provide information about the programming methods used in the present calculations. Information on how to use the programs is provided in Chapter 5, entitled, "User's guide to each of the programs." This chapter, besides the introduction, has four additional sections which describe the field calculation program, the data entry programs, the data structures used in the programs, and, finally, a more detailed description of the programs and their subroutines listed in alphabetical order. This last section also highlights those variables which are important to the understanding of the programs.

The project's purpose is to calculate the maximum electromagnetic field possible in each room of a building from a plane wave impinging directly upon the walls of the building given the dimensions of the building and the materials used in its construction. The program accesses permanent data files which describe the dimensions and material of each layer of each wall, door, and window of the building. These data files are created for each building using separate programs. The program also accesses a previously stored data file which contains the attenuation and reflection coefficients of different construction materials. This data is then used to calculate a transmission coefficient and an absorption coefficient for each wall considering the properties of each layer and each opening. Using these coefficients an energy flow into each room from the outside and from the other rooms is calculated. The energy flowing back out of the building is also then calculated. A steady state energy balance is then assumed for all the rooms, and from this steady state assumption, the energy level in each room is calculated.

4.2 General Description of the Field Calculation Program

A procedure file FIELD gets the main program from mass storage and runs it. The main program MASTER sets up the common blocks and then calls each of the subroutines. Five subroutines (LMATTER, LFREQ, LWALL, LTYPE, and LHOLE) load data from permanent storage into arrays which are accessed by the program when the data is needed for computation. The subroutine LTDB calculates the attenuation and area of each type opening, layer by layer, and loads it into the TDB array.

The CFACTOR subroutine calculates transmission factors of each wall and loads them into the ROOM array. These transmission factors determine the transmission between each room and its adjacent volume, which may be another room or the exterior of the building. The calculation is performed for each material layer of each wall and of each opening in the wall. Note: in this discussion "wall" also means floor and ceiling. These calculations are done sequentially and accumulated as the data for each layer of each opening is accessed in the data files.

The DFACTOR subroutine calculates the absorption factor of each wall and loads them into the DDABS array. In this calculation the absorption is taken as one minus the transmission and minus the reflection. The reflection coefficient is included if a resonance condition is met.

The subroutine ECALC is used to calculate the steady state energy in the rooms. It starts by calling the subroutine SETUP which takes the transmission coefficients from the ROOM arrays and combines them with the absorption coefficients in the DDABS array to produce the proper relationships for energy flow between the rooms and the outside. These linear relationships are put into the TMAP array. The subroutine ECALC then calculates the steady state energy balance in the rooms using the subroutine DETERM. The subroutines RESOND and RESONW calculate resonance conditions for apertures and rooms.

Several subroutines with a "P" prefix print results or file contents. For example, PHOLE, PTYPE, PWALL print the data contained in the data files and the corresponding arrays.

4.3 General Description of the Data Entry Programs

There are five data entry programs: three to store data describing buildings, one to store frequencies, and one to store properties of construction. The three which store the building data are SWALLS, STYPES, and SHOLES; the material properties are stored by SMATDB. The frequencies are stored using SFREQ. These are source code files; the user will only see the equivalent procedure files HSTORE, WSTORE, TSTORE, FSTORE and MSTORE which call and run the compiled binary equivalents of these source files.

All three building description programs use the same interactive format. Each asks the user for a building identifier which can be as many as five characters long. That building identifier is used to create the names for the permanent data files needed for each building. For example, if the building identifier were "A125" then three permanent files would be named and created: "BA125T", "BA125W", "BA125F", and "BA125H". The suffixes "T", "W", "F", and "H" refer to "Type" data, "Wall" data, "Frequency" data and "Hole" data, respectively.

The material data storage program, also interactive, stores attenuation and reflection coefficient data for different frequencies in a permanent file called MATTER. This program will only be used for adding new construction materials to the material data base.

4.4 Data Structures Used in the Programs

Arrays and Variables:

The following gives a brief description of the variables used in the common blocks.

Hole Variables

HMAX: maximum size of hole array. Initially set at 35.

HOLE (1-HMAX, 1-4): Array containing room and aperture ID.

HOLE (x,1): "Direction" part of room identification.

HOLE (x,2): "From room" part of room identification.

HOLE (x,3): "To room" part of room identification.

HOLE (x,4): "Aperture ID".

HTOT: A numeric variable containing the total number of lines in the "hole" data file.

HERR: A numeric variable used to indicate a file handling error and used in the WARNING and ERROR subroutines to trigger a message to the operator.

Type Variables

TMAX: Maximum size of TYPE and TDIM arrays. Initially set at 35.

TYPE (1-TMAX, 1-4): Character array containing dimensions and material of each layer of each type of door or window.

TYPE (x,1): Identification of type, e.g., "WA" meaning window "A" and "DB" meaning door B.

TYPE (x,2): Material of layer, e.g., "M03" meaning Material #3.

TYPE (x,3): Material of the frame, e.g., "M05" meaning Material #5.

TDIM (1-TMAX, 1-4): Numeric array containing dimensions of each door or window. Used in parallel with TYPE array.

TDIM (x,1): Height of opening in meters.

TDIM (x,2): Width of opening in meters.

TDIM (x,3): Thickness of layer in centimeters.

TDIM (x,4): Distance of opening above floor in meters.

TDB1 (1-TMAX): Character array containing opening identification. Used in parallel with TDB2 array.

TDB1 (x): Opening identification, e.g., "WA".

TDB2 (1-TMAX, 1-2): Numeric array containing attenuation and area of opening. Used in parallel with TDB1 array.

TDB2 (x,1): Attenuation of opening.

TDB2 (x,2): Area of opening in square meters.

TTOT: A numeric variable containing the number of lines in the "Type" data files.

TDBTOT: A numeric variable representing the total number of lines in the TDB1 and TDB2 data arrays.

TERR: A numeric variable used to indicate an error and used in the WARNING and ERROR subroutines to trigger a message to the operator.

Material Variables

MAT (1-100): Contains material identifiers such as "M01", "M02", ... "M99".

MATDESC (1-100): Contains description of each material. (70 characters each).

MFREQ (1-100, 1-7) real array: Contains 7 frequencies at which the property data exists.

MATTEN (1-100, 1-7): Contains 7 attenuation values for each material.

QA (1-100): Contains a quality factor for each material.

MRCOEF (1-100, 1-7): Contains 7 reflection coefficients for each material.

QR (1-100): Contains a reflection coefficient quality factor for each material.

Room Variables

DDABS (1-RMAX + 6, 1-RMAX + 6): This numeric array contains the absorption coefficients of the walls in each room.

ENERGY (1-RMAX): This numeric array contains the results of the energy balance calculations and contains the energy in each room.

POWER (1-6): This numeric array contains the power entering the building from each of the six outside directions: North, East, South, West, Top, Bottom.

RAREA (1-RMAX): Array containing surface area of each room.

RMAX: This numeric variable sets the maximum number of rooms that the program can handle. Initially set for 20 for simplicity of the display subroutine when printing on 80 column paper.

ROOM (1-RMAX + 6, 1-RMAX + 6): This numeric array is used for the transmission factors between room and room, and between room and the outside world.

TMAT (1-RMAX + 6, 1-RMAX + 6): This numeric array is the matrix that contains the linear relationships of energy flow between the rooms. It is a combination of information from the ROOM array and the DDABS array, and is created using the subroutine SETUP.

Wall Variables

WMAX: Maximum size of wall arrays. Initially set to 75.

WALL (1-WMAX, 1-4): Character array containing wall identifiers and material identification.

WALL (x,1): Direction--one of three wall identifiers.

WALL (x,2): From room--one of three wall identifiers.

Wall (x,3): To room--one of three wall identifiers.

WDIM (1-WMAX, 1-3): Numeric array containing wall dimensions.

WDIM (x,1): Height in meters.

WDIM (x,2): Width in meters.

WDIM (x,3): Layer thickness in centimeters.

WTOT: A numeric variable containing the number of lines in the Wall data files.

WERR: A numeric variable used to indicate a file handling error and used in the WARNING and ERROR subroutines to trigger a message to the operator.

Miscellaneous Variables

BLDG: This character variable contains the building identifier. It is combined with other strings to identify the various data files associated with that building.

DREFL: This variable contains the reflection coefficient for a wall. If the frequency is not near a room resonance, DREFL is set to zero.

DREFLW: This variable is used to calculate window input resonances. If the frequency is near a window resonance then DREFLW is set to 20 dB. Otherwise, DREFLW is zero.

FMAX: Maximum number of frequency values in FREQA array. Initially set at 50.

FREQ: Frequency in hertz.

FREQA (1-FMAX): A numeric array containing frequencies for calculations.

AFLAG: A number between 0 and 100 which determines how much of the quality factor is applied to the attenuation value.

RFLAG: A number between 0 and 100 which determines how much of the quality factor is applied to the reflection coefficient value.

Labeled Common Blocks:

This section lists variables transmitted by each labeled common block along with the subroutines using the block.

INITILC: BLDG

Common to MASTER, CFACTOR, LHOLE, LTDB, LTYPE, LWALL, PROOM, PPWR, PTMAT, DFACTOR, PDDABS, PPWR2, SPWR, RESONW, RESOND, LFREQ

INITILN: FREQ, QUALITY, AFLAG, RFLAG, FREQA (FMAX), FERR, FTOT

Common to MASTER, CFACTOR, LHOLE, LTDB, LTYPE, LWALL, PROOM, PPWR, RESOND, RESONW, PPWR2, PDDABS, PTMAT, DFACTOR, SPWR, LFREQ

HOLEC: HOLE (HMAX, 4)

Common to MASTER, CFACTOR, LHOLE, DFACTOR, PHOLE

HOLEN: HTOT, HERR

Common to MASTER, CFACTOR, LHOLE, DFACTOR, PHOLE

MATC: MAT (MMAX), MATDESC (MMAX)

Common to MASTER, ATTEN, LMATTER, RCOEF, RESONW

MAT: TMAT (RMAX, RMAX), ENERGY (RMAX), POWER (6), FTIME, SWR (RMAX, 6), IDIR

Common to MASTER, SETUP, ECALC, PPWR, PTMAT, PPWR2, SPWR, PDDABS

MATN: MATTEN (MMAX, 7), MRCOEF (MMAX, 7), QA (MMAX), QR (MMAX), MFREQ (MMAX, 7), MERR, MTOT

Common to MASTER, ATTEN, LMATTER, RCOEF, RESONW

ROOMD: DDABS (RMAX + 6, RMAX + 6), DREFL, DREFLW

Common to MASTER, CFACTOR, SETUP, DFACTOR, LDDABS, IDDABS, RESOND, RESONW

ROOMN: ROOM (RMAX + 6, RMAX + 6), NROOMS, RAREA (RMAX)

Common to MASTER, CFACTOR, LRAREA, LROOM, PROOM, SETUP, ECALC, PPWR, PTMAT, IDDABS,

DFACTOR, LDDABS, PPWR2, SPWR, RESOND, RESONW, PDDABS

TYPEC: TYPE (TMAX,3), TDB1 (TMAX)

Common to MASTER, CFACTOR, LTDB, LTYPE, PTDB, PTYPE, SRCHTDB, DFACTOR, RESOND

TYPEN: TDIM (TMAX, 4), TTOT, TDB2 (TMAX, 2), TDBTOT, TERR

Common to MASTER, CFACTOR, LTDB, LTYPE, PTDB, PTYPE, SRCHTDB, DFACTOR, RESOND

WALLC: WALL (WMAX, 4)

Common to MASTER, CFACTOR, LRAREA, LWALL, PWALL, DFACTOR, RESONW

WALLN: WDIM (WMAX, 3), WTOT, WERR

Common to MASTER, CFACTOR, LRAREA, LWALL, PWALL, DFACTOR, RESONW

4.5 Description of Programs and Subroutines Used in Field Calculation (Listed Alphabetically)

Important variables (not including those passed by common blocks) are listed after each routine.

Subroutines with arguments are listed with the arguments in parentheses.

*FUNCTION ATTEN (ID, FREQ, AFLAG): This real function returns the material attenuation for a specified frequency and for a specified quality. It obtains the attenuation values from the MATTEN array and interpolates according to the frequency.

ID: Material identification such as "M01".

*SUBROUTINE CFACTOR: This subroutine calculates the attenuation of each wall and each opening in each wall, layer by layer, and then calculates transmission factors for each wall.

DREFLW: This value is 20 dB near a window frequency resonance.

WATTEN: Wall attenuation.

OATTEN: Opening attenuation, whether the opening is a door or window.

LATTEN: Layer attenuation.

MATTEN: Material attenuation.

MATT: Material identifier such as "M01".

ID: Identifier of opening.

WALL: Wall array containing Wall identification and Material identification.

*FUNCTION DETERM (ARRAY, NORDER): This real function calculates the determinant of a matrix.

ARRAY (1-RMAX + 6, 1-RMAX + 6): This real array represents the input matrix. It is destroyed during the calculation.

NORDER This real variable represents the order of the input matrix.

*SUBROUTINE DFACTOR: This subroutine calculates the attenuation of each wall and each opening in each wall, layer by layer, and then calculates the absorption factor for each wall.

DREFL: Wall reflection coefficient.

WATTEN: Wall attenuation.

OATTEN: Opening attenuation, whether the opening is a door or a window.

LATTEN: Layer attenuation.

MATTEN: Material attenuation.

ID: Identifier of opening.

WALL: Wall array containing Wall identification and Material identification.

*SUBROUTINE ECALC: This subroutine calculates the energy balance in the rooms. It calls subroutines SETUP and DETERM.

PVECTOR (1-RMAX): This real array contains the values representing the initial power levels injected into each room from the outside field.

*SUBROUTINE ERROR (IERR): This subroutine returns an error message when called with an error number as argument. It also stops the program. The error numbers and error messages are listed below:

IERR	MESSAGE
1	Materials data base is empty
2	Frequency is out of range
3	This material is not in data base
4	Denominator is zero
5	File handling error

*FUNCTION GETLEN (STRING): This integer function returns the number of characters in a character string when given the character string as an argument.

*SUBROUTINE IDDABS: This subroutine initializes the DDABS array.

*SUBROUTINE LDDABS: This subroutine loads the DDABS array.

*SUBROUTINE LFREQ: This subroutine loads the array FREQA from the permanent file "BxxxxxP", where "xxxxx" represents the building identifier.

*SUBROUTINE LHOLE: This subroutine loads the material data base from permanent storage into the HOLE array for access by the program.

*SUBROUTINE LRAREA: This subroutine calculates the surface area of each room and inserts it into the RAREA array.

*SUBROUTINE LMATTER: This subroutine loads the material data base from permanent storage into arrays for access by the program.

*SUBROUTINE LROOM (TS, TS2, FROM, TO): This subroutine loads the transmission coefficients TS and TS2 into the appropriate room location in the ROOM array.

*SUBROUTINE LTDB: This subroutine calculates the attenuation and area of each type opening, layer by layer and loads it into the TDB array (Type Data Base).

*SUBROUTINE LTYPE: This subroutine loads arrays TYPE and TDIM with data from permanent file "BxxxxxT", where "xxxxx" is the building identifier.

*SUBROUTINE LWALL: This subroutine loads arrays WALL and WDIM with data from permanent file "BxxxxxW", where "xxxxx" is the building identifier.

*PROGRAM MASTER: This program is the control section which calls each of the subroutines. The program reads in wall, window and door data; calculates transmission coefficients of each wall; stores the transmission coefficients in the ROOM matrix; and calculates the maximum field in each room on a normalized basis. It is called by the user with procedure file FIELD.

*SUBROUTINE PDDABS: This subroutine prints the ROOM matrix.

*SUBROUTINE PTDB: This subroutine prints the array TDB1 and TDB2 giving the area and attenuation of each door and window type.

*SUBROUTINE PHOLE: This subroutine prints the contents of the HOLE array giving the wall location of the doors and windows.

*SUBROUTINE PPWR: This subroutine prints the contents of the ENERGY array and represents the energy values in the rooms.

*SUBROUTINE PPWR2: A second version of PPWR which allows a more efficient format. It uses the output from the subroutine SPWR stored in the array SWR.

*SUBROUTINE PROOM: This subroutine prints the ROOM array giving the transmission factor of each wall.

*SUBROUTINE PTMAT: This subroutine prints the contents of the TMAT array. It can be used for debugging the program.

*SUBROUTINE PTYPE: This subroutine prints the contents of the arrays TYPE and TDIM giving the parameters of each door and window type.

*SUBROUTINE PWALL: This subroutine prints the contents of the WALL and WDIM arrays.

*SUBROUTINE RESOND (ID): This subroutine calculates the range of frequencies that correspond to a resonance condition for a window or door with a metal frame. If the frequency of the incoming radiation is in the range of the resonance then DREFLW is set to 20 dB. This is used in the CFACTOR subroutine to increase the transmission through a window by 20 dB if resonance occurs.

DREFLW: This is the return variable. (Used in common block ROOMD.)

FLOW: This is the lower frequency bound for resonance.

FHIGH: This is the upper frequency bound for resonance.

ID: This is the identification label for the window.

*SUBROUTINE RESONW (FROM, MATID): This subroutine calculates the frequency range that corresponds to a resonance condition in a room. If a wall has a reflection coefficient greater than 0.80 and the frequency corresponds to resonance for the room, then DREFL is set to the reflection coefficient of the wall under question. Otherwise DREFL is set to zero.

DREFL: This is the return variable. (Used in common block ROOMD.)

FLOW: This is the lower frequency bound for resonance.

FHIGH: This is the upper frequency bound for resonance.

FROM: This identifies the room being calculated.

MATID: This represents the material identification label for the wall being calculated.

*FUNCTION RCOEF (MATID, FREQ, RFLAG): This function returns the material reflection coefficient for a specified frequency and for a specified quality.

*SUBROUTINE SETUP: This subroutine loads the TMAT array.

*SUBROUTINE SPWR: This subroutine saves the energy values as they are calculated so that they can be formatted neatly when printed.

*SUBROUTINE SRCHTDB (ID, OATTEN, OAREA): This subroutine searches the TDB array given the ID label of a door or window and returns the attenuation and area of that door or window.

*FUNCTION VAL (String): This function when given a number expressed as a character string returns the number expressed as an integer.

*SUBROUTINE WARNING (IERR): This subroutine returns an error message when called with a warning number as argument. The warning number and message follows:

IERR	MESSAGE
1	HOLE data file does not exist for this bldg.
2	File handling problem on HOLE data file.
3	MATTER file does not exist for this bldg.
4	File handling problem on MATTER data file.
5	TYPE data file does not exist for this bldg.
6	File handling problem on TYPE file.
7	WALL data file does not exist for this bldg.
8	File handling problem on WALL file.
9	Height and width of room missing.
10	Length of room missing.
11	Frequency file does not exist for this building.
12	File handling problem with FREQ file.

5. USERS' GUIDE TO COMPUTER PROGRAMS FOR DATA ENTRY AND COMPUTATION

5.1 Introduction

The programs discussed in this chapter have been written for use on the Control Data Corporation Cyber 750 computer at the Boulder, Colorado Laboratories of the U.S. Department of Commerce. Though the programs contain checks and tests to help assure their correct use, we urge each user to read this guide carefully and to enter data (e.g., window and door types and locations) in the exact form and sequence which we specify. The checks and tests make the programs somewhat "user-friendly" but not entirely fool-proof (no offense intended).

When an electromagnetic wave is normally incident on an outside wall of a building, we compute the power attenuation of the wave as it penetrates the building by a procedure comprising the data files BxxxxxF, MATTER, BxxxxxW, BxxxxxT, and BxxxxxH, and the program MASTER.

- BxxxxxF: a file containing the frequencies to be used in the calculation. The suffix "F" stands for Frequency. The "xxxxx" in the name represents the identification name of a building, e.g., B90023F would be the frequency data file for building number 90023. This convention is used for all the other data files.
- MATTER: a file containing our computed reflection coefficients and attenuation values for building materials. Users will have direct interaction with this file only if they wish to change data or enter additional data for a material already in the file, or if they wish to enter data for an additional material.
- BxxxxxW: the user enters data on the location, size, and composition of walls in the building to be evaluated. The suffix "W" stands for Walls data.
- BxxxxxT: for each door and window type, the user enters material, size, and a two-character identification. The suffix "T" stands for Types data.
- BxxxxxH: in this file, the user specifies which doors and windows are located in each wall, identifying the door and window types by their two-character codes. The suffix "H" stands for Holes data.
- MASTER: this program computes the power attenuation when an electromagnetic field, incident on an exterior wall, penetrates into any room of a building. MASTER consults the files MATTER, BxxxxxW, BxxxxxT, and BxxxxxH to obtain the material and building data necessary for the computation. It uses the file BxxxxxF to determine which frequencies to use.

All programs are written in FORTRAN V for use on the CDC 750 computer.

5.2 Data Preparation for Programs SWALLS, STYPES, and SHOLES

The sequence ending in an attenuation computation begins with the user drawing a plan of the building to be analyzed. This plan helps the user derive the specifications for walls, doors, and windows which are then entered into the data files BxxxxxW, BxxxxxT, and BxxxxxH.

At this point we discuss typical steps in reducing the floor plan in figure 5.1 to a set of specifications acceptable to MASTER. The procedures we illustrate with this simple example can be used in the same manner for more complex structures. The only restriction on shape is that the floor plan be rectangular or composed of adjoining rectangles. The same restriction applies to room shapes. Examine the building thoroughly to include all features (walls, doors, windows) that determine its shielding characteristics.

The building plan must be labeled as follows (see the example in fig. 5.1):

1. D1, D2, D3, D4 denote the exterior regions, or "directions", around the building. These regions must be labeled in order to specify from which direction the radiation is coming. If necessary, D5 and D6 can be the regions above and below the building, or above and below a room (e.g., a second-floor room).
2. The directions LR (left-to-right) and FB (front-to-back) specify which walls are parallel to each other. This information is useful if two parallel walls (of the same room) have high reflection coefficients, because the region between them may contain the intensified fields of standing waves produced by reflections between the walls. The program MASTER computes the highest and lowest frequencies at which these resonances may occur.
3. Label the rooms $\emptyset 1$, $\emptyset 2$, $\emptyset 3$. . .
4. Determine the window and door types in the building; label these WA, WB, ..., and DA, DB, ..., respectively, at their locations on the floor plan.

To prepare data for entry into the file BxxxxxW, the user should make a data sheet to specify the size, orientation, and composition of the walls. A suggested format is shown in figure 5.2:

1. The number of each line of data is given in the "LINE#/" column. When changing or entering a line of data, the user employs the line number. Note that the line numbers are shown in Figure 5.1, also.
2. The column "DIRECTION" specifies the direction to which the wall in a given line is perpendicular. This direction must be the same as that defined by the "FROM" and "TO" columns.
3. Note that lines 1, 2, 3 specify the material layers in the wall between region D4 and room $\emptyset 1$, and that these layers are encountered in that sequence in going from D4 to $\emptyset 1$. The material layers in a wall must be entered in the file in the correct sequence corresponding to the direction given by the "FROM" and "TO" columns.

4. The wall "HEIGHT" (meters), "LENGTH" (meters), and "THICKNESS" of material layer (centimeters) are entered in their respective columns.
5. The material identification number of each layer is entered in the "MATERIAL" column according to the material data tables in chapter 2.

We strongly advise that the user employ some means of marking the building plan as each wall is entered into the BxxxxxW data table. When the table is complete and all walls are so marked (e.g., a pencil check; a colored highlight), the user will know that none have been omitted (or perhaps entered twice). It can also be helpful to put the data table line numbers onto the drawing as they are entered into the table. In figure 5.1, these line numbers are shown at the intersections of the walls with the cross-sections (dashed lines) through the building.

To tabulate the types of doors (D) and windows (W) indicated in figure 5.1, we suggest the format in figure 5.3. From left to right, the columns specify doors and windows as follows:

1. Line numbers (the "LINE" column) are used in adding, deleting, and displaying data.
2. The floor plan in figure 5.1 has door types DA, DB, DC, and window types WA, WB.
3. A door or window has a height of H meters and a width of W meters, where H and W are the inside dimensions of a door frame and a window sash (the frame in which the glass is set).
4. "DISTANCE ABOVE FLOOR" is the distance in meters from the floor to the bottom edge of the glass or screen in a window.
5. In the next two columns, the "LAYER MATERIAL" (door or window) and its "THICKNESS" (in cm) are specified for each door and window. Notice that doors and windows may contain more than one material: window type WA has galvanized mesh screen (material M09) and window glass (M08). The user may also encounter storm windows, windows with thermopane (double-layer) glass, storm doors, and screen doors.
6. "FRAME MATERIAL"; obtain identification (e.g., M04) for the frame material from the material data tables.

In the specification of walls and the openings in them, all that remains now is to prepare data for BxxxxxH, the holes file which tells MASTER where the doors and windows are located. A suitable format for tabulating this data is given in figure 5.4. As before, a wall is identified by specifying 1) the two regions between which the wall is located and 2) the direction to which the wall is perpendicular (this orients the wall with respect to left-right or front-back directions). Because the wall "from D4 to 01" has two types of openings, it is listed twice (lines 3 and 4).

We again urge users to prepare their data for entry into the computer files by using the tabular formats we have suggested (figures 5.2, 5.3, 5.4). This procedure will reduce careless errors in transcribing building specifications from the floor plan to files BxxxxxW, BxxxxxT, and BxxxxxH.

5.3 How to Become a Remote-Site Time-Share User

Having prepared a floor plan and building data tables, the user must now enter this data into the files BxxxxxF, BxxxxxW, BxxxxxT, and BxxxxxH. Before we present details of how this is done, we describe how one becomes an off-site time-share user of the Control Data Corporation (CDC) 170/750 computer system at the Boulder Laboratories of the U.S. Department of Commerce.

To establish a relationship with the computer, call User Services: (303) 497-5849 or (303) 497-5850 (on FTS, 303 is a direct dial area code). When first contacting the Computer Services Division through User Services, prospective users must furnish 1) their name, organization name, and telephone number, and 2) the name and telephone number of an "authority contact" (i.e., a project leader, supervisor, etc.). User Services will then provide a user number and an initial password. When a billing account is established and the way is cleared for use of the computer, the user changes the initial password to another one which is then secure, known only to the user.

A password (4-7 alphanumeric characters) must be changed every three months, otherwise the computer will assign a new one known only to the computer. The user will then be unable to get on line without first going through User Services to submit a new password. At each log-in for about ten days prior to the expiration of a password, the computer reminds the user to enter a new one. A password is changed by the command

PASSWOR, OLDPSWD, NEWPSWD

in which the user supplies the old password (OLDPSWD) and then the new password (NEWPSWD).

To obtain an account number for computer charges, the user's purchasing department should send a purchase order to

Ms. Beverly Armstrong
NOAA/ERL, R-E52
U.S. Department of Commerce
325 Broadway
Boulder, Colorado 80303.

Ms. Armstrong can be reached on (303) 497-5842 (FTS direct dial).

Dan Smith (303-497-5846) or Ron Buxton (303-497-5845) can advise which direct-dial extension to use to match the baud rate on the user's terminal.

5.4 The CDC 170/750: Data Entry and Computation

5.4.1 Log-In Procedure

Once the user is on-line, the initial conversation with the computer will be as shown in figure 5.5. In reply to the first three requests, the user enters: family name, user number, and password. If no family name is required, press the CR key (CR = carriage return). When typed, the user number does not appear on the printout. The password is typed over the blackened area. After the word "CHARGE:", the user enters

CHARGE, Z1234567,Z

(or a similar form specified by User Services) where "1234567" represents the user's account number. As seen in this printout, entering the word BYE disconnects the user from the computer.

5.4.2 Procedure File MSTORE

The file MATTER contains all the attenuation and reflection data in the tables in chapter 2. To add or change data, one uses the procedure file MSTORE (M for material), and obtains it with the BEGIN command

BEGIN, MSTORE

Figure 5.6 illustrates how the user enters building material data by means of MSTORE, which first asks if the user wants to (1) create a new data base, or (2) add to the existing data base. The user makes the choice by typing 1 or 2 after the question mark. (Note that whenever MSTORE awaits a user reply, it types a question mark as a prompt to the user.) If a data base is already present in the MATTER file and the user chooses to create a new data base, typing 1 ERASES THE EXISTING BASE. Before making this choice, the user must be certain this is indeed what is wanted. Also, one must take care not to inadvertently type the numeral 1 when 2 is intended.

The entry of material description and data is complete when the user has responded to the six prompts. (NOTE: a character string must be typed within single quotes; e.g., 'MOIST CLAY BRICK', 'M15'.) MSTORE then presents the choice of (1) adding data for another material, (2) changing data in the base, (3) displaying the data for material in the base, (4) canceling (aborting) the data set just entered, or (5) quitting the data-entry procedure. The choice (5) enters the new data into the file MATTER.

Figure 5.7 shows the user how to use MSTORE to change data already in MATTER. After the material identification M05 is entered, MSTORE prints all specifications for M05, and the user then enters the line number of the data to be changed. The procedure to change the attenuation quality percent from 100 to 10 is then self-evident. After all changes have been made and the user has entered 99 to leave the "change" mode, the complete revised data for M05 is presented. Entering 5 (the "quit" option)

disconnects the user from MSTORE and enters the revised data into the file MATTER. To reestablish contact with the procedure file MSTORE, the user must again enter BEGIN, MSTORE.

The entire data entry for a material can be deleted simply by using the "change" option to replace the material identification in line 1 (e.g., M05) with a blank. Enter two consecutive single quotes as the changed material ID; the absence of information between them is the blank.

5.4.3 Data File BxxxxxW (W for Walls)

In the file BxxxxxW, the user stores the location, dimensions, and composition of each wall in each building to be analyzed. Every building thus specified in BxxxxxW must be identified by a string of no more than five alphanumeric characters represented by xxxxx here. Building identification is the first information requested as the user begins entering or changing data in the BxxxxxW file.

Manipulating data in BxxxxxW is done through the procedure file WSTORE; that is, WSTORE is the procedure file by which the user creates and/or alters the data file BxxxxxW. Contact with BxxxxxW is therefore initiated by the command

BEGIN, WSTORE

Figure 5.8(a) illustrates the format for entering data, via WSTORE, into the file BxxxxxW. First, the user enters an identification number assigned to the building whose walls are to be documented. This identification number replaces the "xxxxx" in the file name forming a unique name. Then comes the payoff for care taken in tabulating wall data (fig. 5.2), for WSTORE now requests this data in the left-to-right order of the tabulation. Note that when data has been entered for a wall layer, entering another layer for that wall requires only "thickness of layer" and "material ID" (directional data, height, and width are the same for all layers in a given wall). If the user discontinues data entry, WSTORE presents the seven options at the top of figure 5.8(b).

Option 1: a line can be displayed as shown in figure 5.8(b). Entering the number "0" instead of a line number allows the user to leave the display mode.

Option 2: options 2 and 5 are similar in that they both involve adding data lines to an existing file. While 5 is only for adding a data line at the end of the file, option 2 allows the user to insert a data line between two other lines (fig. 5.8(c)). When this is done, the inserted line must be part of a wall already represented in the file (otherwise, option 5 would be used). For this reason, the user must be sure that the directions, height, and width of the inserted line match those of one (or perhaps both) of the adjacent lines (depending on whether one or both of the adjacent lines belong to the door/window of the inserted line). If the user enters the data incorrectly and the line does not match at least one of its neighbors, WSTORE will emphatically point out the error. This error message is shown at the bottom of figure 5.8(c). Note that the faulty line has

not been accepted (i.e., not entered into the BxxxxxW files); the user must again choose option 2 and enter the corrected data.

Option 3: this option allows the user to delete a line of data (fig. 5.8(d)). The procedure is self-explanatory. Revised data can be entered using option 2 or 5, whichever is appropriate.

Option 4: displays all data for this building so far entered into BxxxxxW.

Option 5: adds a line of data at the end of an existing file. If the added line is another layer in the last wall in the file, the user is asked only for layer thickness and material ID. If the added line is the first layer of a new wall in the file, the format for entering the data is the same as in option 2 (fig. 5.8(c)). Neither choice within option 5 requires matching the entered data with that in an adjacent line.

Options 6 and 7: option 6 (fig. 5.8(d)) stores new data in file BxxxxxW, or replaces existing data in BxxxxxW with a revised version of that data. If the user does not wish to store the new or revised data just entered, option 7 can be used to cancel that data and leave the BxxxxxW file unchanged.

5.4.4 Data File BxxxxxT (T for Types)

In this file the user stores data from the table of door and window types and specifications (fig. 5.3). The procedure for entering this data is TSTORE, and the procedures for data entry, the cautionary comments, and the error messages are almost identical to those employed in the procedure WSTORE. Therefore, familiarity with WSTORE is sufficient warmup for using the procedure TSTORE to enter data into the file BxxxxxT. Figure 5.9 illustrates the format for data entry using TSTORE; the similarity with WSTORE is obvious.

Because doors and windows may have layers (e.g., storm doors, storm windows, screens), an insert-line-into-file option is again one of seven data-handling choices available. As in WSTORE, an inserted line of data represents an additional layer of a door or window already in the file. Therefore, the identifier (e.g., DE, WC), frame material, height, width, and distance above floor in the inserted line must match those specifications in one or both of the adjacent lines. If not, the error message and the procedure for changing the incorrect line of data are the same as in WSTORE.

The user gains access to TSTORE with the command

BEGIN, TSTORE.

Building identification is required before data entry begins.

5.4.5 Data File BxxxxxH (H for Holes)

The user stores in the file BxxxxxH the types and locations of all doors and windows in the building to be analyzed; location is given by specifying the wall containing each door and window. This data

should have been previously tabulated as in figure 5.4, and is now to be entered into the data file BxxxxxH by means of the procedure HSTORE. Again, the summons is

BEGIN, HSTORE

and building identification is required to initiate data entry (fig. 5.10).

Though not necessary, it may be a bookkeeping convenience to enter together the data for all doors and windows in a given wall. If option 2 ("insert line into file") is employed for this purpose, the user will be pleased to know that HSTORE does not require matching between inserted and adjacent lines. Therefore, HSTORE, while very similar to MSTORE, WSTORE, and TSTORE, is also simpler and will present no difficulties to a user familiar with the other three data entry programs.

5.4.6 Data File BxxxxxF (F for Frequency)

The user stores in the file BxxxxxF the frequencies required for the calculation using the procedure file FSTORE. The program is begun by the command,

BEGIN, FSTORE.

If no BxxxxxF file is created, the program MASTER will use a set of default frequencies in the calculation.

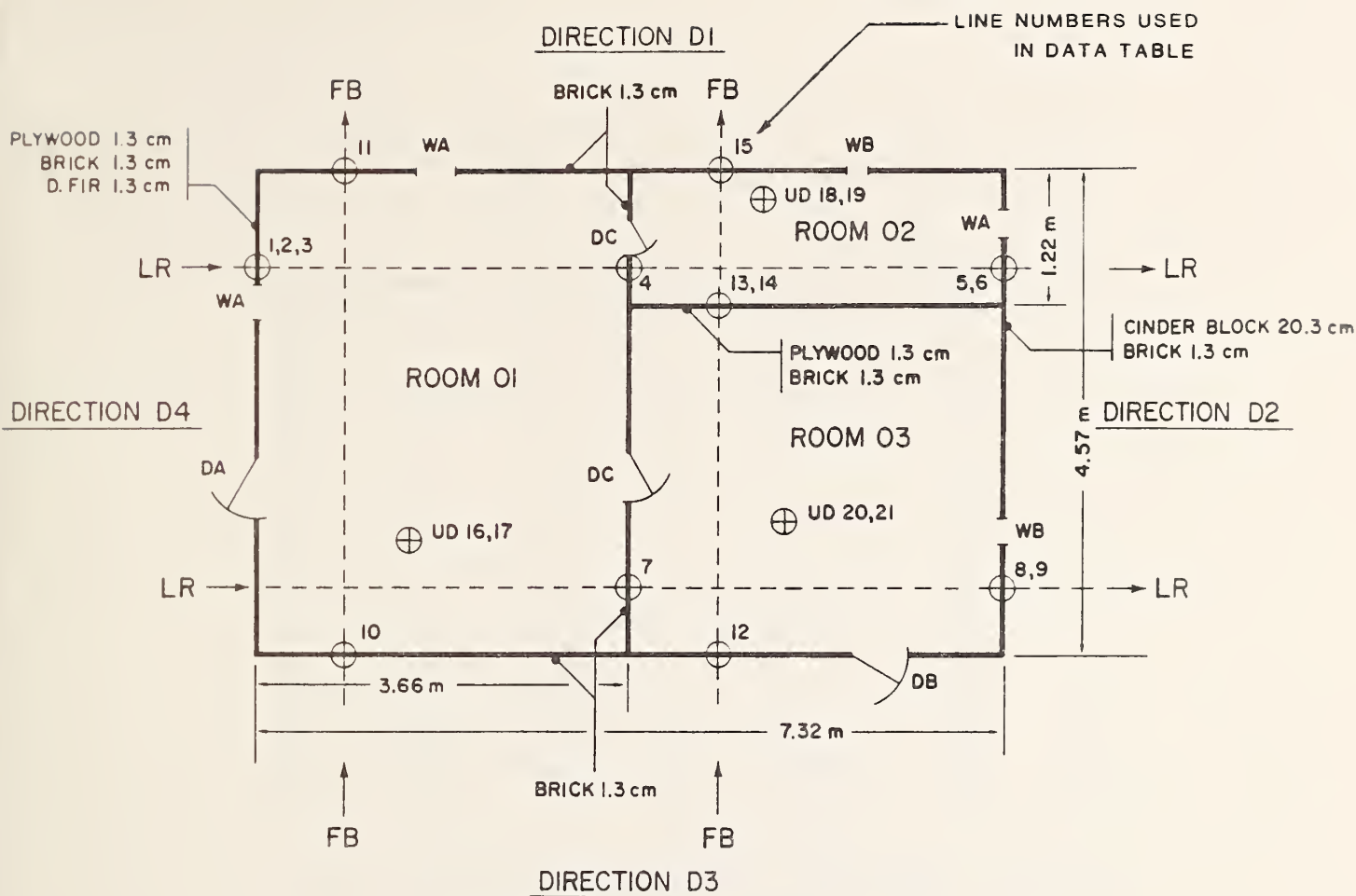
5.4.7 Computation Program MASTER

The program MASTER obtains building data from the files MATTER, BxxxxxW, BxxxxxT, and BxxxxxH, and computes the attenuation (in dB) for each room; that is, the attenuation of externally incident radiation as it penetrates into each room of the building. The user must be sure these files contain all the necessary data before consulting MASTER. When the files are ready, the user activates MASTER by entering

BEGIN, FIELD.

The procedure file FIELD summons MASTER, the data files, and the subroutines required by MASTER for its data handling operations. The only information that FIELD asks of the user is the identification code of the building to be analyzed. When the user enters this building identification, the computation begins.

For each of the five directions from which radiation may be incident on the building (e.g., north, east, south, west, above), MASTER computes all room attenuations for the frequencies given in the data file BxxxxxF. If the frequency file is missing, MASTER will use the seven default frequencies given in the material data tables. The printed output begins with a listing of the data files BxxxxxW, BxxxxxH, and BxxxxxT (fig. 5.11a); then the attenuation values are given (fig. 5.11b) by frequency, direction, and room. The data in figure 5.11 a,b are for the sample building whose wall, door, and window specifications were obtained from figure 5.1. Actual results based on field measurements are shown in chapter 7.



NOTES:

WINDOW DATA

- WA 1.2 m (H) x 0.91 m (W), plywood frame, 1 cm glass with 0.61 cm thick 24x24 galvanized steel mesh
- WB 1.2 m (H) x 1.5 m (W), aluminum frame, 1 cm glass with 0.61 cm thick 24x24 galvanized steel mesh

DOORS

- DA 2.10 m (H) x 0.91 m (W) plywood door, 5.1 cm thick with wood frame
- DB 2.10 m (H) x 1.20 m (W) plywood door 5.1 cm thick with wood frame
- DC 2.10 m (H) x 2.10 m (W) plywood door 1.3 cm thick with aluminum frame

WALLS

All heights are 2.44 m.

FLOOR

Cement 1.30 cm thick.

CEILING

Wood (Fir) 1.30 cm

Figure 5.1 Layout of sample building used for illustrating data input.

LINE #	DIRECTION	FROM	TO	HEIGHT	WIDTH	THICKNESS	MATERIAL
1	LR	D4	01	2.44	4.57	1.30	M05
2	LR	D4	01	2.44	4.57	1.30	M06
3	LR	D4	01	2.44	4.57	1.30	M04
4	LR	01	02	2.44	1.22	1.30	M06
5	LR	02	D2	2.44	1.22	20.30	M07
6	LR	02	D2	2.44	1.22	1.30	M06
7	LR	01	03	2.44	3.35	1.30	M06
8	LR	03	D2	2.44	3.35	20.30	M07
9	LR	03	D2	2.44	3.35	1.30	M06
10	FB	D3	01	2.00	3.66	1.30	M06
11	FB	01	D1	2.44	3.66	1.30	M06
12	FB	D3	03	2.44	3.66	1.30	M06
13	FB	03	02	2.44	3.66	1.30	M05
14	FB	03	02	2.44	3.66	1.30	M06
15	FB	02	D1	2.44	3.66	1.30	M06
16	UD	D5	01	4.57	3.66	1.30	M04
17	UD	01	D6	4.57	3.66	1.30	M02
18	UD	D5	02	1.22	3.66	1.30	M04
19	UD	02	D6	1.22	3.66	1.30	M02
20	UD	D5	03	3.35	3.66	1.30	M04
21	UD	03	D6	3.35	3.66	1.30	M02

Figure 5.2. Example of wall data tabulation for walls in figure 5.1.

LINE	ID	HEIGHT (METERS)	WIDTH (METERS)	DISTANCE ABOVE FLOOR	THICKNESS (CM)	LAYER MATERIAL	FRAME MATERIAL
1	DA	2.10	.91	0.00	5.10	M05	M05
2	DB	2.10	1.20	0.00	5.10	M05	M05
3	DC	2.10	2.10	0.00	1.30	M05	M13
4	WA	1.20	.91	.61	1.00	M08	M05
5	WA	1.20	.91	.61	.61	M09	M05
6	WB	1.20	1.50	.91	1.00	M08	M13
7	WB	1.20	1.50	.91	.61	M09	M13

Figure 5.3. Format for preparing door and window specifications for entry into file BxxxxxT.

LINE #	DIRECTION	FROM	TO	ID
1	LR	03	D2	WB
2	FB	D3	03	DB
3	LR	D4	01	WA
4	LR	D4	01	DA
5	LR	01	02	DC
6	LR	01	03	DC
7	FB	01	D1	WA
8	FB	02	D1	WB
9	LR	02	D2	WA

Figure 5.4. Format for preparing door and window location data for entry into file BxxxxxH (sample data from figure 5.1).

```

83/09/14. 14.14.30.
N O A A / M A S C 170/750 83/06/26. NOS 1.4 531/552.40
FAMILY:
    USER NUMBER:
                PASSWORD
XXXXXXXXXX
TERMINAL:      220, TTY
RECOVER/ CHARGE: CHARGE,Z7233491,Z
/bye

WYSS          LOG OFF    14.14.49.
WYSS          SRU        1.002 UNITS.

```

Figure 5.5. The log-in procedure.

```

/BEGIN,MSTORE.
(1) CREATE NEW DATABASE (2) ADD TO EXISTING DATA BASE
? 2
(1) NEXT DATA ENTRY (2) CHANGE (3) DISPLAY (4) ABORT (5) QUIT
? 1

MATERIAL I.D.? (E.G. M05 OR M12)
? 'M15'
INDEX:15
ENTER ONE LINE DESCRIPTION OF MATERIAL
? 'MOIST CLAY BRICK'
ENTER 7 ATTENUATION VALUES FROM LOW TO HIGH FREQ
? .0000022,.0000102,.00014,.0025,.00572,.00572,.0057
ENTER ATTENUATION QUALITY PERCENT
? 10.
ENTER 7 REFLECTION COEFFS FROM LOW TO HIGH FREQ
? .13,.072,.051,.029,.014,.014,.014
ENTER REFLECTION COEFFICIENT QUALITY PERCENT
? 100.

(1) NEXT DATA ENTRY (2) CHANGE (3) DISPLAY (4) ABORT (5) QUIT
? 5

REVERT. MSTORE COMPLETED.
/

```

Figure 5.6. Entering building material data into the file MATTER.

```

/8EGIN,MSTORE.
(1) CREATE NEW DATABASE (2) ADD TO EXISTING DATA BASE
? 2
(1) NEXT DATA ENTRY (2) CHANGE (3) DISPLAY (4) ABORT (5) QUIT
? 2

ENTER MATERIAL I D. OF GROUP TO BE CHANGED
? 'M05'

LINE 1 MATERIAL ID
M04
LINE 2 DESCRIPTION
FIR PLYWOOD
LINE 3 FREQUENCIES
10000. 100000. 1000000. 10000000. 100000000. 1000000000.
1 E+10
LINE 4 ATTENUATIONS
.00000115 .00000677 .00000831 .000124 .00215 .026 .3
LINE 5 ATTENUATION QUALITY PERCENT
100.
LINE 6 REFLECTION COEFFICIENTS
.068 .048 .074 .036 .014 .013 .01
LINE 7 REFLECTION QUALITY PERCENT
100.

ENTER NUMBER OF LINE TO BE CHANGED (99 TO END CHANGES)
? 5
ENTER NEW ATTENUATION QUALITY PERCENT
? 10.

ENTER NUMBER OF LINE TO BE CHANGED (99 TO END CHANGES)
? 99

M05
FIR PLYWOOD
10000. 100000. 1000000. 10000000. 100000000. 1000000000.
1.E+10
.00000115 .00000677 .00000831 .000124 .00215 .026 .3
10.
.068 .048 .074 .036 .014 .013 .01
100.

(1) NEXT DATA ENTRY (2) CHANGE (3) DISPLAY (4) ABORT (5) QUIT
? 5

REVERT. MSTORE COMPLETED.
/

```

Figure 5.7. Changing building material data in the file MATTER by means of the procedure file MSTORE.

```

/BEGIN,WSTORE.

ENTER BUILDING IDENTIFICATION (E.G. '101')
(NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'

WILL THIS BE
(1) A MODIFICATION OF AN EXISTING FILE?
(2) A NEW FILE?
ENTER A NUMBER !!!
? 2

BEGIN ENTERING DATA

ENTER DIRECTION (E. G. 'LR')
? 'LR'

ENTER "FROM" (E.G. '02' OR 'D1')
? 'D4'

ENTER "TO" (E.G. '02' OR 'D1')
? '01'

ENTER HEIGHT, METERS
? 2.44

ENTER WIDTH, METERS
? 4.57

ENTER THICKNESS OF LAYER, CENTIMETERS
? 1.30

ENTER "MATERIAL ID" (E.G. 'M01')
? 'M05'

DO YOU WANT TO ENTER MORE DATA?(1) YES (2) NO
ENTER A NUMBER !!!
? 1

IS THIS THE FIRST LAYER OF A WALL (1) YES (2) NO
ENTER "0" TO ESCAPE "DATA ENTRY" MODE
ENTER A NUMBER!!
? 2

ENTER THICKNESS OF LAYER, CENTIMETERS
? 1.30

ENTER "MATERIAL ID" (E.G. 'M01')
? 'M06'

DO YOU WANT TO ENTER MORE DATA?(1) YES (2) NO
ENTER A NUMBER !!!
? 2

DATA ENTRY DISCONTINUED

```

Figure 5.8(a). Procedure file WSTORE is used to enter wall data into file BxxxxxW. Data is entered in the sequence of columns in figure 5.2.

```

CHOOSE
(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA

ENTER A NUMBER !!!

? 1

SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED
( ENTER "0" TO ESCAPE DISPLAY MODE )

? 1

LINE #   DIRECTION   FROM   TO   HEIGHT   WIDTH   THICKNESS   MATERIAL
  1         LR        D4    01    2.44    4.57    1.30        M05

CHOOSE
(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA

ENTER A NUMBER !!!

? 6

DOUBLE CHECK !!!
DO YOU YOU WANT TO STORE THIS DATA AND END PROG
NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE
OF THE SAME NAME !!!
ENTER A NUMBER: (1) YES (2) NO

? 1

DATA HAS BEEN STORED AND PROGRAM TERMINATED
REVERT. WSTORE COMPLETED.
/

```

Figure 5.8(b). (Continued from figure 5.8(a)) How the display mode allows the user to examine a line of data.


```

/BEGIN,WSTORE.

ENTER BUILDING IDENTIFICATION (E.G. '101')
(NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'

WILL THIS BE
(1) A MODIFICATON OF AN EXISTING FILE?
(2) A NEW FILE?
ENTER A NUMBER !!!
? 1

CHOOSE
(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA
ENTER A NUMBER !!!
? 2

SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS TO BE INSERTED
( ENTER "0" TO ESCAPE "INSERTION" MODE )
? 2

ENTER DIRECTION (E. G. 'LR')
? 'LR'

ENTER "FROM" (E.G. '02' OR 'D1')
? '06'

ENTER "TO" (E.G. '02' OR 'D1')
? '05'

ENTER HEIGHT, METERS
? 2.44

ENTER WIDTH, METERS
? 2.0

ENTER THICKNESS OF LAYER, CENTIMETERS
? 2.0

ENTER "MATERIAL ID" (E.G. 'M01')
? 'M01'

YOUR DATA WAS NOT ACCEPTED !!!
YOUR DATA MUST REPRESENT A LAYER IN AN EXISTING WALL
I.E. THE DIRECTION, FROM, TO, HEIGHT, AND WIDTH
PARAMETERS MUST MATCH THE WALL JUST BEFORE
OR JUST AFTER YOUR SPECIFIED INSERTION POINT

THE FOLLOWING DISPLAYS
THE LINE BEFORE YOUR LINE,
YOUR LINE, AND THE LINE AFTER

```

LINE #	DIRECTION	FROM	TO	HEIGHT	WIDTH	THICKNESS	MATERIAL
1	LR	D4	01	2.44	4.57	1.30	M05
2	LR	06	05	2.44	2.00	2.00	M01
3	LR	D4	01	2.44	4.57	1.30	M06

Figure 5.8(c). The WSTORE sequence for inserting data into the file BxxxxxW. Note error message when an incorrect line is inserted.

```

CHOOSE
(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA

```

ENTER A NUMBER !!!

? 3

```

SPECIFY THE NUMBER OF THE LINE TO BE DELETED
(ENTER "0" TO ESCAPE DELETION MODE)

```

? 1

DOUBLE CHECK !!!

DO YOU WANT TO DELETE THE FOLLOWING LINE?:

LINE #	DIRECTION	FROM	TO	HEIGHT	WIDTH	THICKNESS	MATERIAL
1	LR	D4	01	2.44	4.57	1.30	M05

ENTER (1) YES OR (2) NO

? 1

LINE # 1 DELETED

CHOOSE

```

(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA

```

ENTER A NUMBER !!!

? 6

DOUBLE CHECK !!!

DO YOU YOU WANT TO STORE THIS DATA AND END PROG

NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE
OF THE SAME NAME !!!

ENTER A NUMBER: (1) YES (2) NO

? 1

DATA HAS BEEN STORED AND PROGRAM TERMINATED

REVERT. WSTORE COMPLETED.

/

Figure 5.8(d). WSTORE: how to delete data (option 3), and use of option 6 to store entered data and terminate WSTORE.

```

/BEGIN, TSTORE

ENTER BUILDING IDENTIFICATION (E.G. '101')
(NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? 701
701 <-ERROR IN COL. 4, RETYPE RECORD FROM THIS FIELD
? '701'

WILL THIS BE
(1) A MODIFICATION OF AN EXISTING FILE?
(2) A NEW FILE?
ENTER A NUMBER !!!
? 2

BEGIN ENTERING DATA

ENTER 'ID' (E.G. 'WA' OR 'DE')
? 'WA'

ENTER HEIGHT, METERS
? 2

ENTER WIDTH, METERS
? .98

ENTER DISTANCE ABOVE FLOOR, METERS
? 1

ENTER THICKNESS OF LAYER, CENTIMETERS
? .4

ENTER "MATERIAL ID OF LAYER" (E.G. 'M01')
? 'M05'

ENTER "MATERIAL ID OF FRAME" (E.G. 'M01')
? 'M08'

DO YOU WANT TO ENTER MORE DATA?(1) YES (2) NO
ENTER A NUMBER !!!
? 1

IS THIS THE FIRST LAYER OF A DOOR OR WINDOW? (1) YES (2) NO
ENTER "0" TO ESCAPE "DATA ENTRY" MODE
ENTER A NUMBER!!
? 2

ENTER THICKNESS OF LAYER, CENTIMETERS
? 1

ENTER "MATERIAL ID OF LAYER" (E.G. 'M01')
? 'M04'

DO YOU WANT TO ENTER MORE DATA?(1) YES (2) NO
ENTER A NUMBER !!!
? 2

DATA ENTRY DISCONTINUED

CHOOSE
(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA
ENTER A NUMBER !!!

```

Figure 5.9. The procedure TSTORE enters data on door and window types into file BxxxxxT.

```

/BEGIN,HSTORE

ENTER BUILDING IDENTIFICATION (E.G. '101')
(NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'

WILL THIS BE
(1) A MODIFICATION OF AN EXISTING FILE?
(2) A NEW FILE?
ENTER A NUMBER !!!
? 2

BEGIN ENTERING DATA

ENTER DIRECTION (E. G. 'LR')
? 'LR'

ENTER "FROM" (E.G. '02' OR 'D1')
? 'D2'

ENTER "TO" (E.G. '02' OR 'D1')
? '03'

ENTER HOLE 'ID' (E.G. 'WA' OR 'DA')
? 'WB'

DO YOU WANT TO ENTER MORE DATA?(1) YES (2) NO
ENTER A NUMBER !!!
? 1

ENTER DIRECTION (E. G. 'LR')
? 'TB'
DIRECTION MUST BE 'LR' OR 'FB' OR 'UD'
TRY AGAIN!!!

ENTER DIRECTION (E. G. 'LR')
? 'FB'

ENTER "FROM" (E.G. '02' OR 'D1')
? 'D3'

ENTER "TO" (E.G. '02' OR 'D1')
? '03'

ENTER HOLE 'ID' (E.G. 'WA' OR 'DA')
? 'DB'

DO YOU WANT TO ENTER MORE DATA?(1) YES (2) NO
ENTER A NUMBER !!!
? 2

DATA ENTRY DISCONTINUED

CHOOSE
(1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES
(2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
(3) DELETE LINE (6) STORE DATA AND EXIT PROGRAM
(7) EXIT PROGRAM WITHOUT STORING DATA
ENTER A NUMBER !!!

? 4

LINE # DIRECTION FROM TO ID
1 LR D2 03 WB
2 FB D3 03 DB

```

Figure 5.10. Procedure HSTORE enters into file BxxxxxH the types and locations of doors and windows.

```

/BEGIN.FIELD
ENTER BUILDING IDENTIFICATION (E.G. '101')
      (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'
BUILDING IDENTIFICATION ENTERED AS '701'
ENTER NUMBER OF ROOMS IN BUILDING
? 3

```

WALL IDENTIFICATION				WALL PARAMETERS		
DIR	FROM	TO	MATERIAL	HEIGHT	WIDTH	THICKNESS
LR	D4	01	M05	2.44	4.57	1.30
LR	D4	01	M06	2.44	4.57	1.30
LR	D4	01	M04	2.44	4.57	1.30
LR	01	02	M06	2.44	1.22	1.30
LR	02	D2	M07	2.44	1.22	20.30
LR	02	D2	M06	2.44	1.22	1.30
LR	01	03	M06	2.44	3.35	1.30
LR	03	D2	M07	2.44	3.35	20.30
LR	03	D2	M06	2.44	3.35	1.30
FB	D3	01	M06	2.00	3.66	1.30
FB	01	D1	M06	2.44	3.66	1.30
FB	D3	03	M06	2.44	3.66	1.30
FB	03	02	M05	2.44	3.66	1.30
FB	03	02	M06	2.44	3.66	1.30
FB	02	D1	M06	2.44	3.66	1.30
UD	D5	01	M04	4.57	3.66	1.30
UD	01	D6	M02	4.57	3.66	1.30
UD	D5	02	M04	1.22	3.66	1.30
UD	02	D6	M02	1.22	3.66	1.30
UD	D5	03	M04	3.35	3.66	1.30
UD	03	D6	M02	3.35	3.66	1.30

DOOR AND WINDOW LOCATIONS

WALL IDENTIFICATION

ID	DIRECTION	FROM	TO
WB	LR	03	D2
DB	FB	D3	03
WA	LR	D4	01
DA	LR	D4	01
DC	LR	01	02
DC	LR	01	03
WA	FB	01	D1
WB	FB	02	D1
WA	LR	02	D2

DOOR AND WINDOW PARAMETERS

ID	MATERIAL	FRAME MATERIAL	HEIGHT	WIDTH	LAYER THICKNESS	DISTANCE ABOVE FLR
DA	M05	M05	2.10	.91	5.10	0.00
DB	M05	M05	2.10	1.20	5.10	0.00
DC	M05	M13	2.10	2.10	1.30	0.00
WA	M08	M05	1.20	.91	1.00	.61
WA	M09	M05	1.20	.91	.61	.61
WB	M08	M13	1.20	1.50	1.00	.91
WB	M09	M13	1.20	1.50	.61	.91

Figure 5.11(a). An output of the program MASTER giving room attenuation (in dB) vs. frequency and direction of the incident radiation for building shown in figure 5.1.

ATTENUATION AT A FREQUENCY OF 1.000E+06 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5 *

* 1 *	-.96	-11.21	-1.34	-.41	2.23 *
* 2 *	1.13	-2.92	-5.32	-12.28	.89 *
* 3 *	-3.86	-.51	.65	-11.80	2.70 *

ATTENUATION AT A FREQUENCY OF 1.000E+07 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5 *

* 1 *	-.97	-11.22	-1.35	-.41	2.23 *
* 2 *	1.13	-2.93	-5.32	-12.28	.89 *
* 3 *	-3.87	-.52	.65	-11.80	2.70 *

ATTENUATION AT A FREQUENCY OF 1.000E+08 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5 *

* 1 *	-.37	-3.52	-.42	-2.09	2.83 *
* 2 *	-2.92	-6.51	-4.03	-5.75	-.45 *
* 3 *	-1.98	-3.33	-.78	-3.75	2.07 *

ATTENUATION AT A FREQUENCY OF 1.000E+09 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5 *

* 1 *	-.97	-12.41	-1.36	-.49	2.20 *
* 2 *	1.12	-4.11	-5.40	-12.40	.84 *
* 3 *	-3.92	-1.69	.60	-11.91	2.66 *

ATTENUATION AT A FREQUENCY OF 1.000E+10 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5 *

* 1 *	-1.02	-23.00	-1.42	-1.24	1.90 *
* 2 *	1.07	-14.45	-6.10	-13.56	.41 *
* 3 *	-4.30	-12.02	.21	-13.00	2.30 *

REVERT. FIELD COMPLETED.

/

Figure 5.11(b). Continuation of figure 5.11(a).

6. EXPERIMENTAL DATA

This chapter comprises reports on building attenuation measurements made by NBS at the Seneca Falls Army Depot, the Sierra Army Depot, and the Naval Training Equipment Center.

6.1 Sierra and Seneca Falls Army Depots

Equipment and personnel were assembled to make measurements of electromagnetic attenuation of three buildings at two sites selected by the Army. These sites were the Sierra Army Depot in Herlong, California, and the Seneca Army Depot in Seneca Falls, New York. The measurements were made according to the Draft Test Plan shown in Appendix 9.1, and are briefly described here. The test plan was modified for each test site by the engineers in charge to conform to any time constraints and to any constraints imposed by the physical layout of the structures. For example, the frequency range was reduced from 180 kHz - 18 GHz to 200 kHz - 10 GHz because it was decided that the additional expense involved was not justified by the present needs of the Army. Although the draft test plan recommends four physical group measurements, on the judgment of the engineers in charge, three (or in some cases, two) groups were assumed sufficient to determine the shielding effectiveness of the buildings.

The measurements were conducted over the frequency range of 200 kHz through 10 GHz. The building (No. 672) measured at the Sierra Army Depot was constructed of concrete with a metal barrier inside the wall. Two buildings were measured at the Seneca Falls Army Depot. One (No. 816) was a concrete building with one meter of dirt covering the side of the building and massive steel doors on the ends. The second building (No. 819) was constructed with cement blocks and metal doors for access.

The technique for all the measurements was as follows: a transmitting antenna was placed 30 meters from the building with the transmitted energy being directed perpendicular to the wall of the building. To get a base unattenuated signal the receiving system measured the transmitted signal outside the building at the designated frequency. The same receiving system was then used inside the building to measure the signal attenuation of the building. This signal attenuation (signal inside divided by signal outside, in dB) was measured on a predetermined distance grid within the building. Wherever possible, a $\lambda/4$ (one-quarter wavelength) offset measurement was made to detect the possibilities of building resonance effects. The receiving system used to measure these building attenuations was the isotropic antenna system developed at the National Bureau of Standards.

The attenuation measurements are summarized in tables 6.1 - 6.4 and are plotted in figures 6.1, 6.2, 6.2 and 6.4. The dimensions and physical layouts of the buildings are shown in figures 6.5, 6.6 and 6.7. The tables list the locations, the frequencies of the test, the mean values of attenuation measured, and

uncertainties of the measurement (one standard deviation) shown in parentheses. Computation of the standard deviation is treated in Appendix 9.1.

6.2 Naval Training Equipment Center (NTEC)

Measurements were made of electromagnetic attenuation of three shelters connected together as a training module. The test site was located at NTEC in Orlando, Florida. The measurements were made according to the test plan shown in Appendix 9.1 and are briefly described here. The test plan was modified at the test site by the engineers in charge to conform to any constraints imposed by time or by the physical layout of the test areas.

The measurements were conducted over the frequency range of 35 MHz up to 18 GHz. As shown in figure 6.8, two ground level transmitter locations (labeled 1 and 2) were used. A third location, with the transmitter placed on the roof of an adjacent building, was used to simulate transmissions from above. The following procedure was used for all measurements. The transmitter was placed at one of the locations and was set to operate at the desired frequency and power level listed in the test plan. The receiving antenna was then placed 5 meters outside the building to measure a reference field strength. The receiving antenna was then placed inside Room 1 at the various locations shown in the test plan for the particular frequency. The attenuation (in dB) was then calculated as the measured field strength inside divided by the measured reference field strength outside. In all cases, the receiver electronics were located in Room 2 and were connected to the antennas with electrical cables.

The measured data is shown in tables 6.5 - 6.7 listed by location. The attenuation data is listed as "average", "high", and "low" values for each data point. The first label represents the average of all the data for that location, polarization, frequency, and type of field (magnetic or electric), while the other labels represent the highest and lowest value recorded.

Figures 6.9 - 6.12 show plots of the data where the circles represent electric field attenuation, while the squares represent magnetic field attenuation. The high and low values listed in the tables are shown as horizontal bars above and below each circle or square. When it would be overlapped by the size of the circle or square, the horizontal bar is not shown.

Table 6.1. EM Attenuation of Building No. 672 at Sierra Army Depot

Frequency MHz	Building Attenuation*, dB	
	Electric Field	Magnetic Field
0.2	36 (2)	24 (3)
4	34 (5)	33 (2)
14	27 (4)	
28	33 (4)	
50	39 (6)	
140	37 (5)	
200	27 (4)	
401	22 (4)	
751	29 (3)	
998	28 (3)	
1008	27 (4)	
2000	26 (5)	
4008	39 (5)	
8007	34 (3)	

*Uncertainties representing one standard deviation are shown in parentheses.

Table 6.2. EM Attenuation of Building No. 816, End Wall at Seneca Falls, New York

Frequency MHz	Building Attenuation*, dB	
	Electric Field	Magnetic Field
0.2	46 (6)	33 (14)
4	54 (7)	
15	57 (7)	
30	51 (8)	
50	33 (7)	
100	44 (5)	
200	44 (5)	
400	48 (13)	
750	50 (6)	
1000	45 (7)	
2000	45 (7)	
4000	42 (5)	
8000	55 (8)	

*Uncertainties representing one standard deviation are shown in parentheses.

Table 6.3. EM Attenuation of Building No. 816, Side Wall at Seneca Falls, New York

Frequency MHz	Building Attenuation*, dB	
	Electric Field	Magnetic Field
0.2	56 (2)	50 (9)
4	49 (2)	
15	53 (4)	
30	62 (4)	
50	60 (5)	
100	56 (7)	
500	65 (12)	
1000	63 (7)	
8000	> 83†	
10,000	> 83†	

*Uncertainties representing one standard deviation are shown in parentheses.

†No measurable signal levels with available equipment.

Table 6.4. EM Attenuation of Building No. 819, End Wall at Seneca Falls, New York

Frequency MHz	Building Attenuation*, dB	
	Electric Field	Magnetic Field
0.2	21 (3)	24 (4)
4	18 (7)	16 (5)
15	16 (2)	
30	19 (4)	
50	4 (4)	
100	10 (7)	
200	13 (7)	
500	13 (4)	
750	14 (3)	
1000	23 (4)	
4000	24 (4)	
8000	34 (18)	
10,000	35 (3)	

*Uncertainties representing one standard deviation are shown in parentheses.

Table 6.5. Attenuation measured with launch antenna at location 1, NTEC

Frequency GHz	Vertical Polarization						Horizontal Polarization		
	Electric Field Attenuation (dB)			Magnetic Field Attenuation (dB)			Electric Field Attenuation (dB)		
	Average	High	Low	Average	High	Low	Average	High	Low
0.0035	-34	-43	-25	-38	-43	-34			
0.007	-23	-29	-18	-27	-33	-21			
0.014	-42	-42	-41	-45	-46	-44			
0.028	-43	-49	-37	-43	-46	-40			
0.054	-26	-26	-26						
0.088	-17	-18	-15						
0.14	-15	-16	-13				-16	-18	-14
0.20	-19	-21	-16				-13	-15	-11
0.40	-23	-25	-18				-10	-15	-5
0.75	-20	-23	-19				-31	-31	-30
1.0	-10	-16	-0				-10	-13	-7
2.0	-7	-8	-5				-7	-8	-4
8.0	-37	-38	-35				-23	-26	-20
12.0									
18.0	-11	-11	-11				-13	-15	-12

Table 6.6. Attenuation measured with launch antenna at location 2, NTEC

Frequency GHz	Vertical Polarization						Horizontal Polarization		
	Electric Field Attenuation (dB)			Magnetic Field Attenuation (dB)			Electric Field Attenuation (dB)		
	Average	High	Low	Average	High	Low	Average	High	Low
0.0035	-22.5	-25	-20	-21	-21	-20			
0.007	-30	-40	-21	-20	-20	-19			
0.014	-27	-28	-26	-25	-27	-23			
0.028	-28	-33	-22	-26	-34	-18			
0.054	-32	-39	-25						
0.088	-13	-17	-10						
0.14	- 6	- 8	- 1				-13	-14	-11
0.20	- 7	-10	- 5				-17	-19	-14
0.40	-12	-14	-10				- 7	-11	- 4
0.75	-13	-16	-11				-18	-20	-16
1.0	-22	-24	-21				-20	-26	-17
2.0	-31	-34	-27				-22	-26	-19
8.0	-17	-17	-16				-19	-20	-18
12.0	-13	-24	- 1				-15	-21	- 9
18.0									

Table 6.7. Attenuation measured with launch antenna at location 3 (roof), NTEC

Frequency GHz	Vertical Polarization			Horizontal Polarization		
	Electric Field Attenuation (dB)			Electric Field Attenuation (dB)		
	Average	High	Low	Average	High	Low
8.0	-20	-22	-18	-20	-22	-17

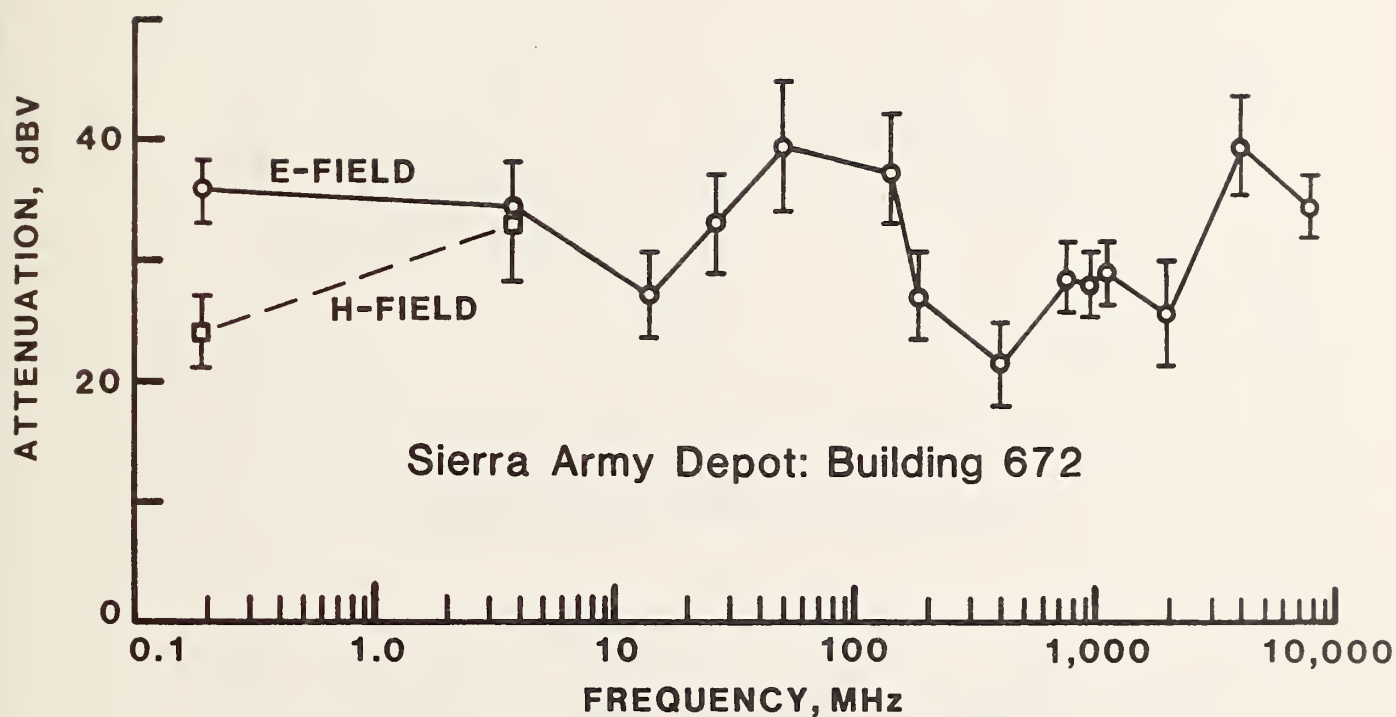


Figure 6.1. Electromagnetic attenuation versus frequency of Building No. 672 at Sierra Army Depot. Electric field attenuation (open circles) was measured from 0.2 - 10,000 MHz while magnetic field (boxes) was measured at 0.2 and 4 MHz. Error bars represent one standard deviation. See figure 6.5 for locations of transmitter and receivers.

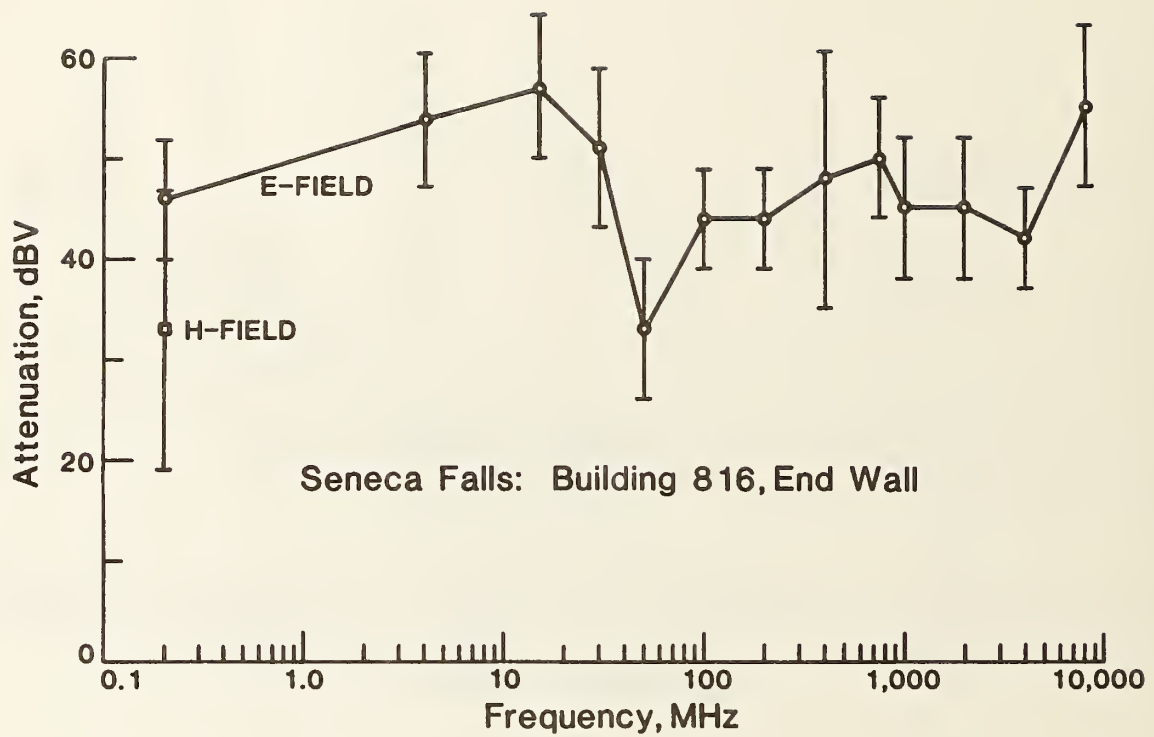


Figure 6.2. Electromagnetic attenuation versus frequency for Building No. 816 at Seneca Falls. The data are plotted as figure 6.1. For this scan, the transmitter was located at the "End Wall" as shown in figure 6.6.

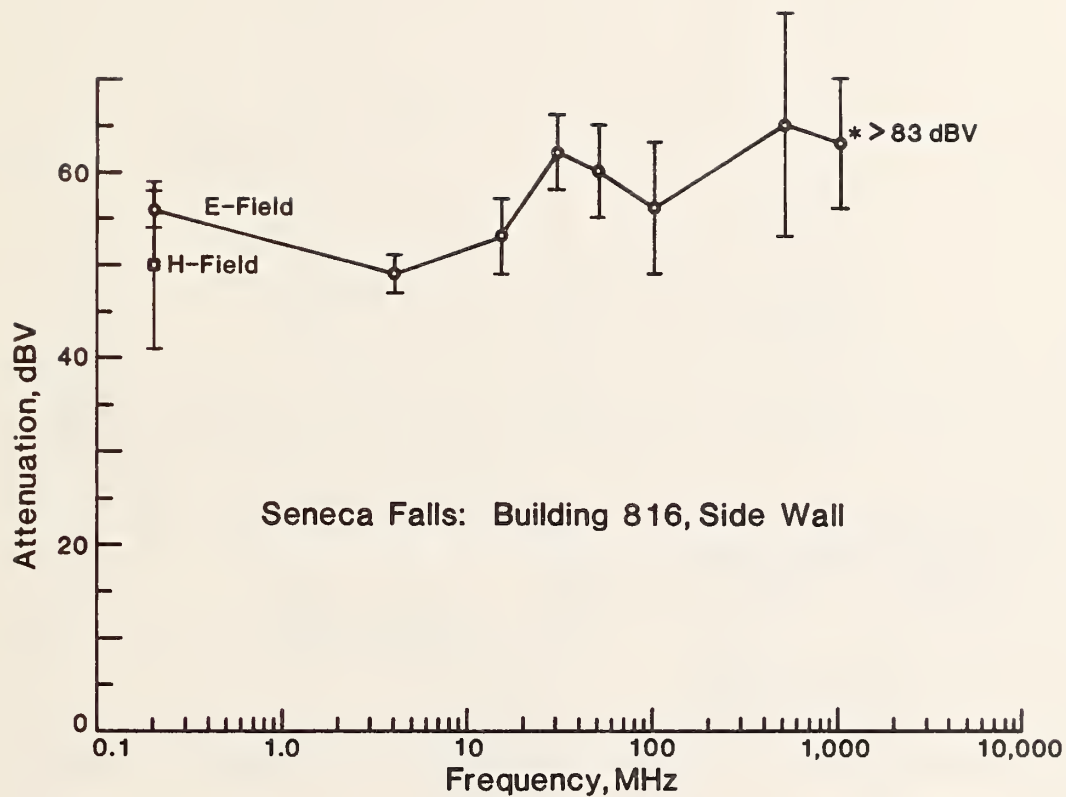


Figure 6.3. Electromagnetic attenuation versus frequency for Building No. 816 at Seneca Falls. The data are plotted as in figure 6.1. For this scan, the transmitter was located at the "Side Wall" as shown in figure 6.6. Magnetic field attenuation was measured at 0.2 MHz, only.

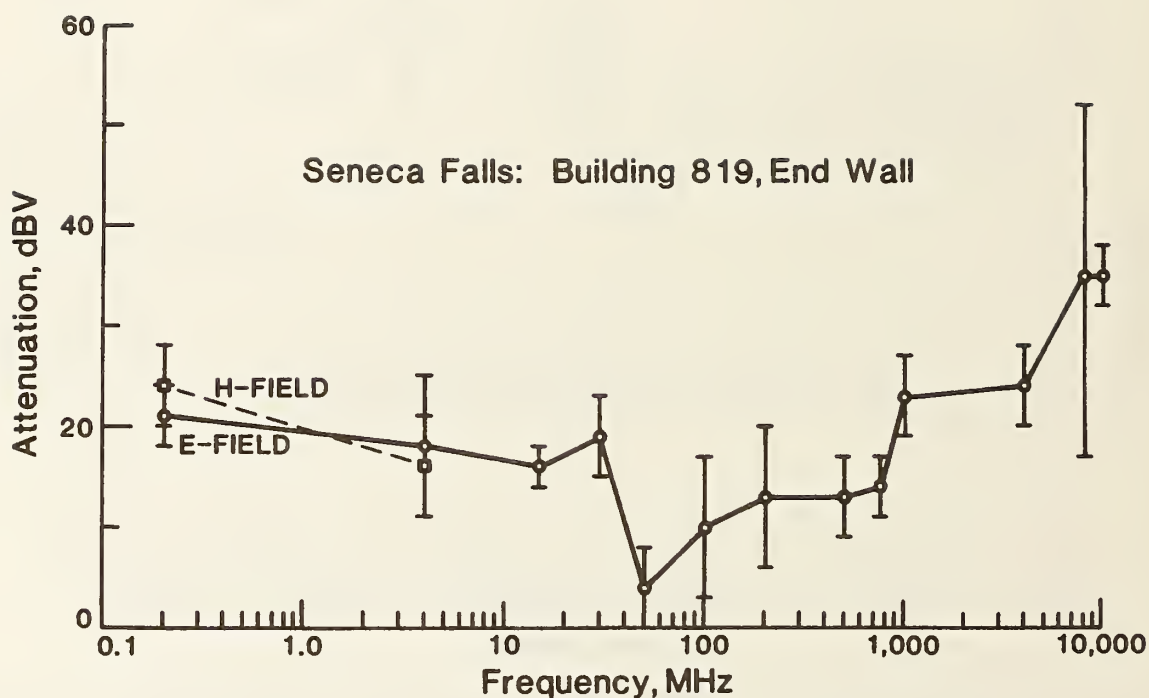


Figure 6.4. Electromagnetic attenuation versus frequency for Building No. 819 at Seneca Falls. The data are plotted the same as in figure 6.1. The transmitter and receiver locations are shown in figure 6.7. Magnetic field attenuation was measured at 0.2 and 4 MHz.

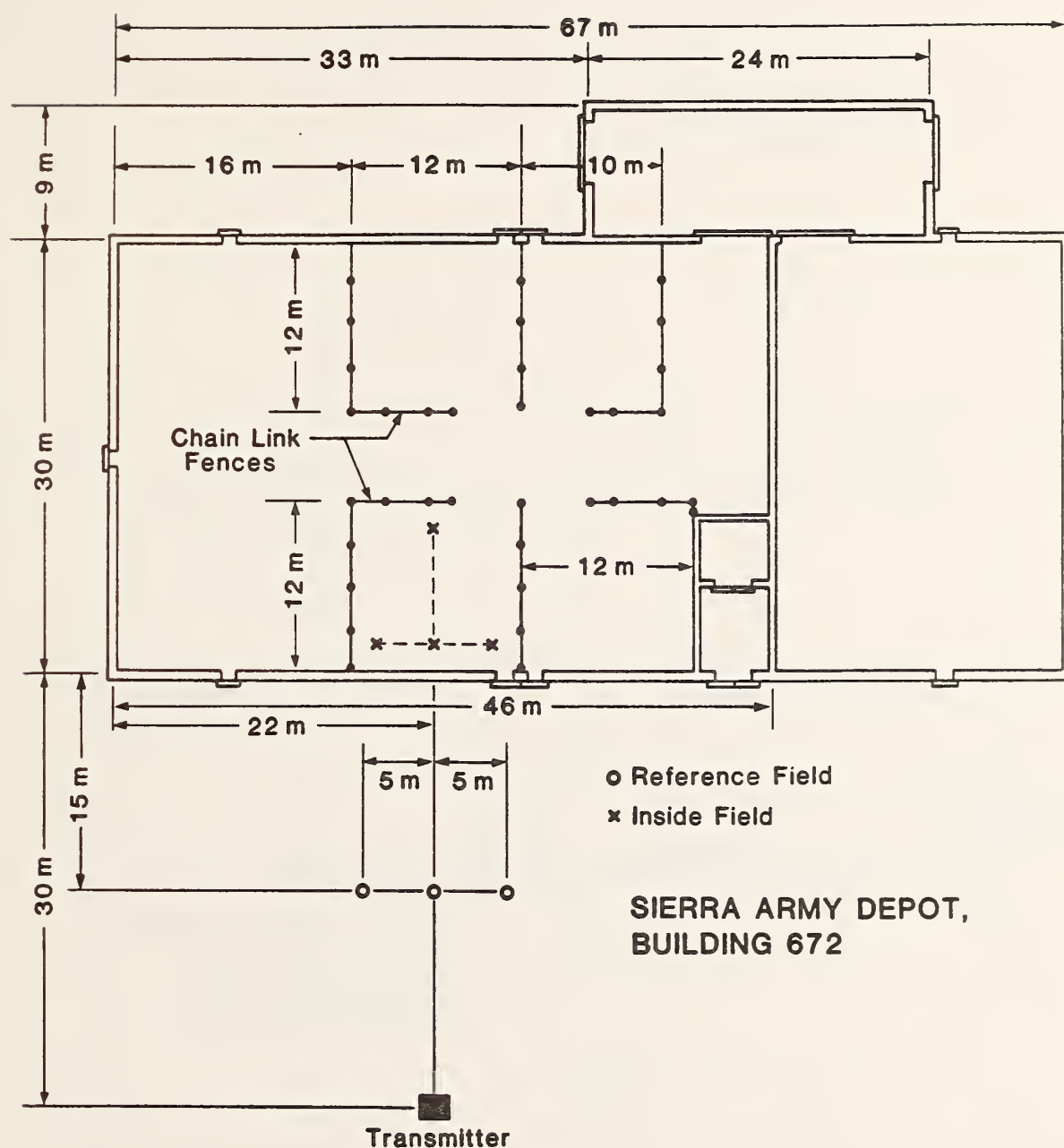


Figure 6.5. Physical layout of Building No. 672 at Sierra Army Depot. The transmitter is located 30 meters from the building and is shown as a square box. Outside reference fields were measured 15 meters from the building at the locations marked with open circles. One row and one column of inside measurements were made at the locations marked by two dashed lines with x's. Measurements are made at one meter intervals along the dashed line.

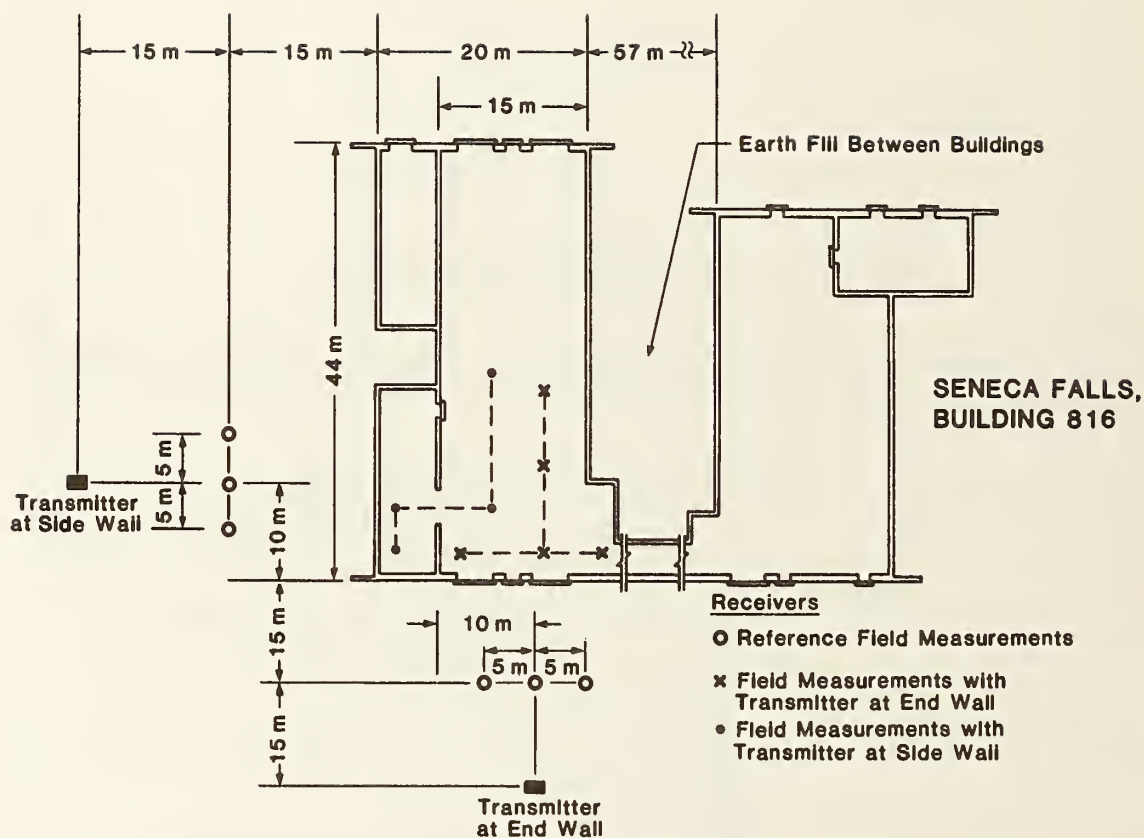


Figure 6.6. Physical layout of Building No. 816 at Seneca Falls. For this site, two sets of data were taken with the transmitter located at the two boxes marked on the drawing. Reference, outside measurements were taken at the locations marked with open circles while inside measurements were taken along the dashed line. For the end wall transmitter location, inside measurements were taken at the locations marked with x's; while the side wall transmitter measurements are marked with closed circles. Measurements are made at one meter intervals along the dashed lines.

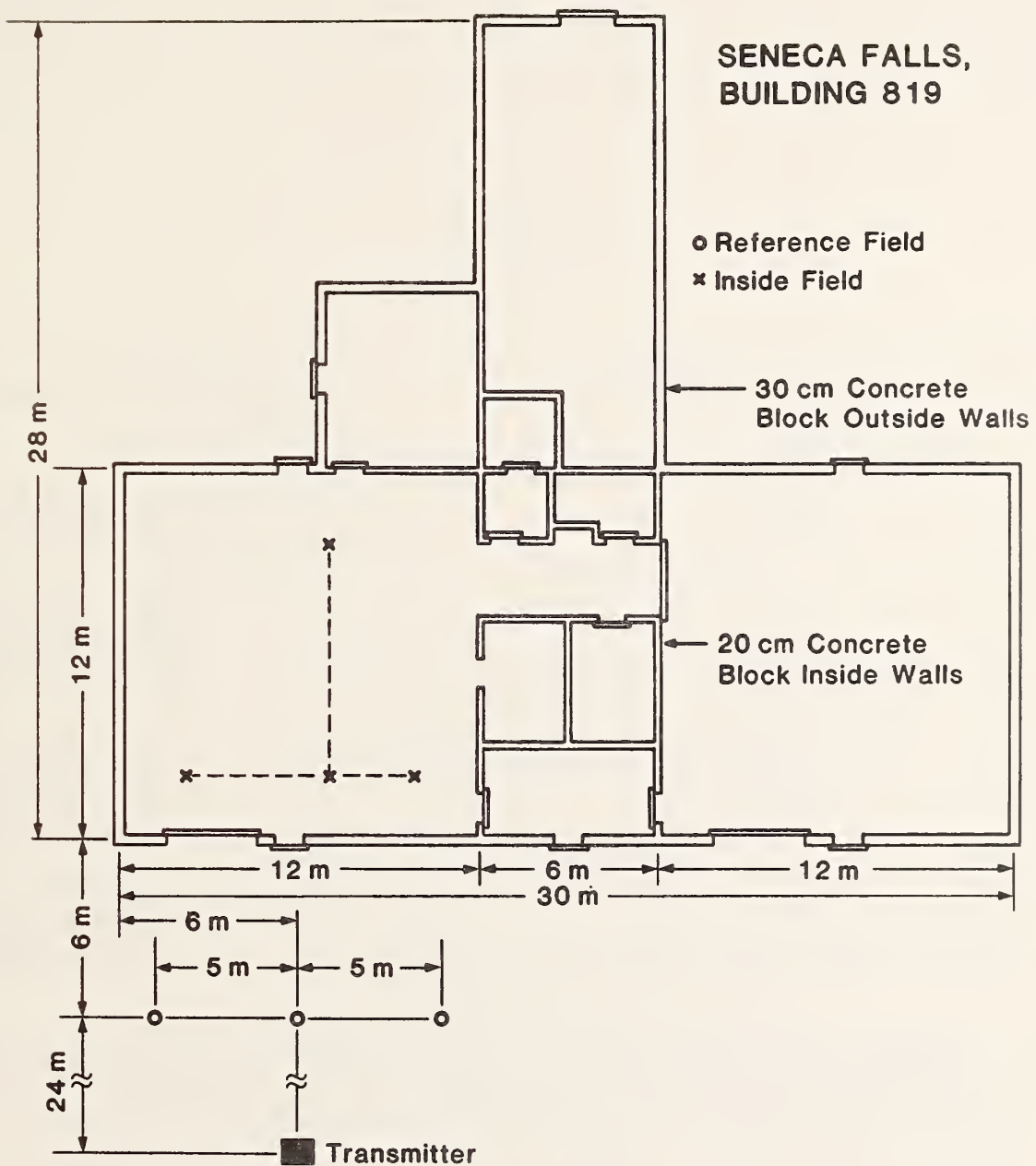


Figure 6.7. Physical layout of Building No. 819 at Seneca Falls. The transmitter is located at the solid box marked on the drawing, while the outside reference points are shown as open circles. The inside measurement points are located along the dashed lines with the x's marked. Measurements are made at one meter intervals along the dashed lines.

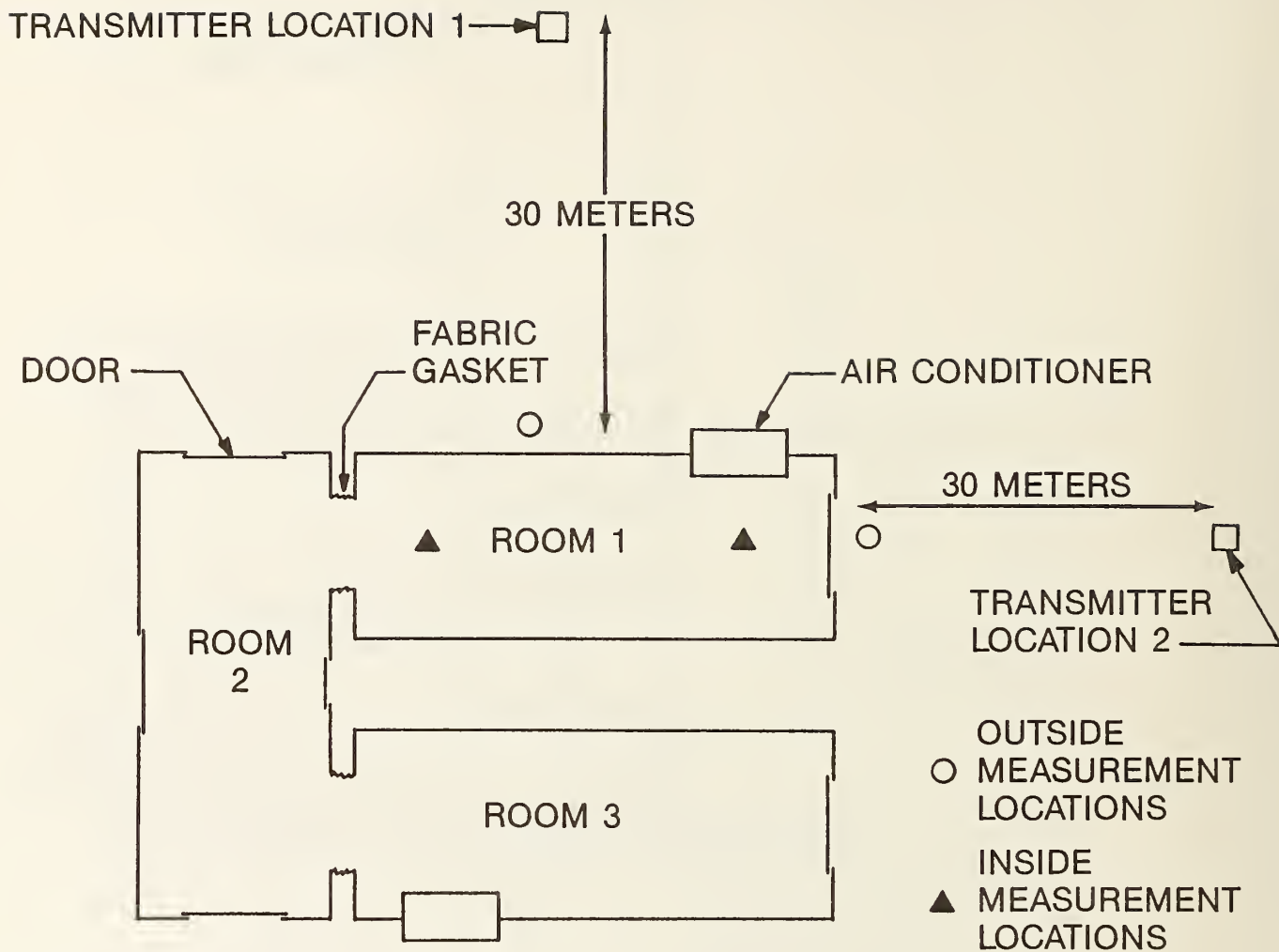


Figure 6.8. Building layout of training module at NTEC. The physical arrangement of the three attached buildings are shown along with the two transmitter locations (boxes). Air conditioning units are labeled "A.C.". A rubber shroud or gasket is used to attach the rooms together.

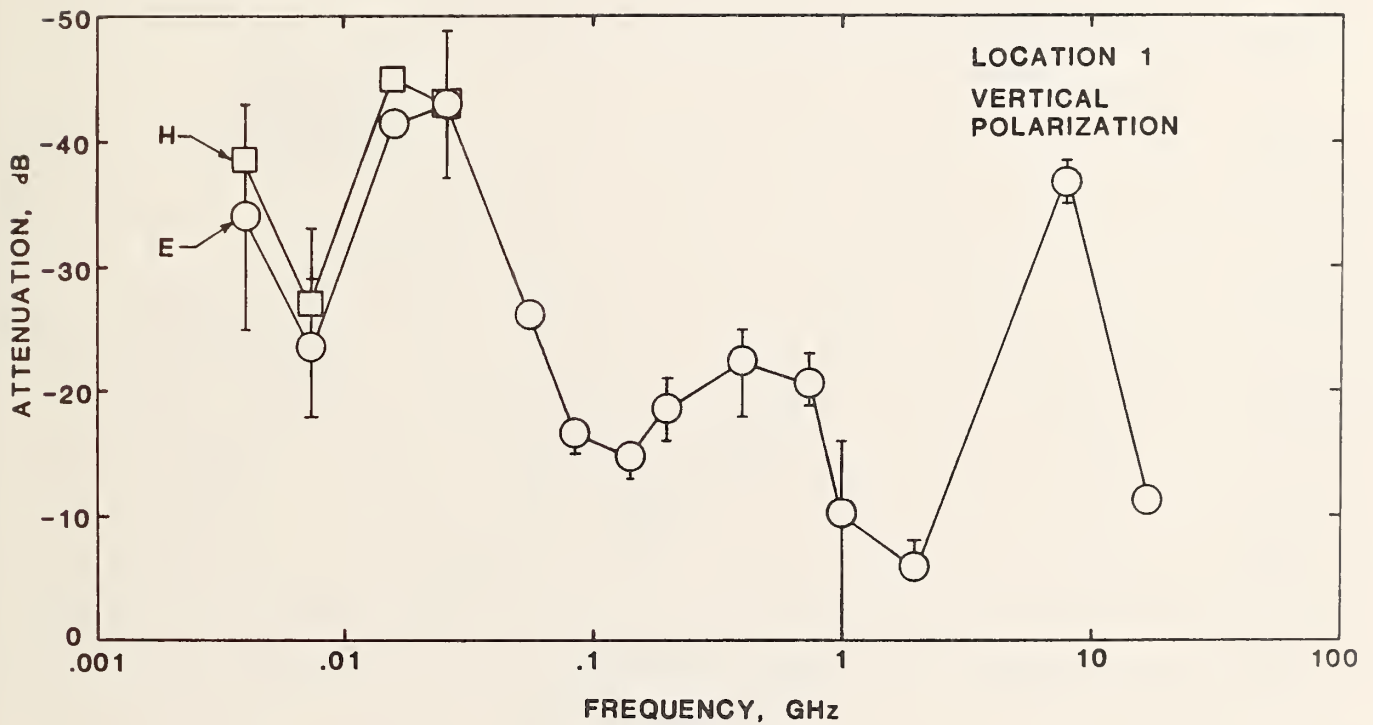


Figure 6.9. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 1 with vertical polarization launched. The circles represent average electric field attenuation, while the average magnetic field attenuation is shown as a square. The limit bars represent the highest and lowest attenuation observed at each frequency.

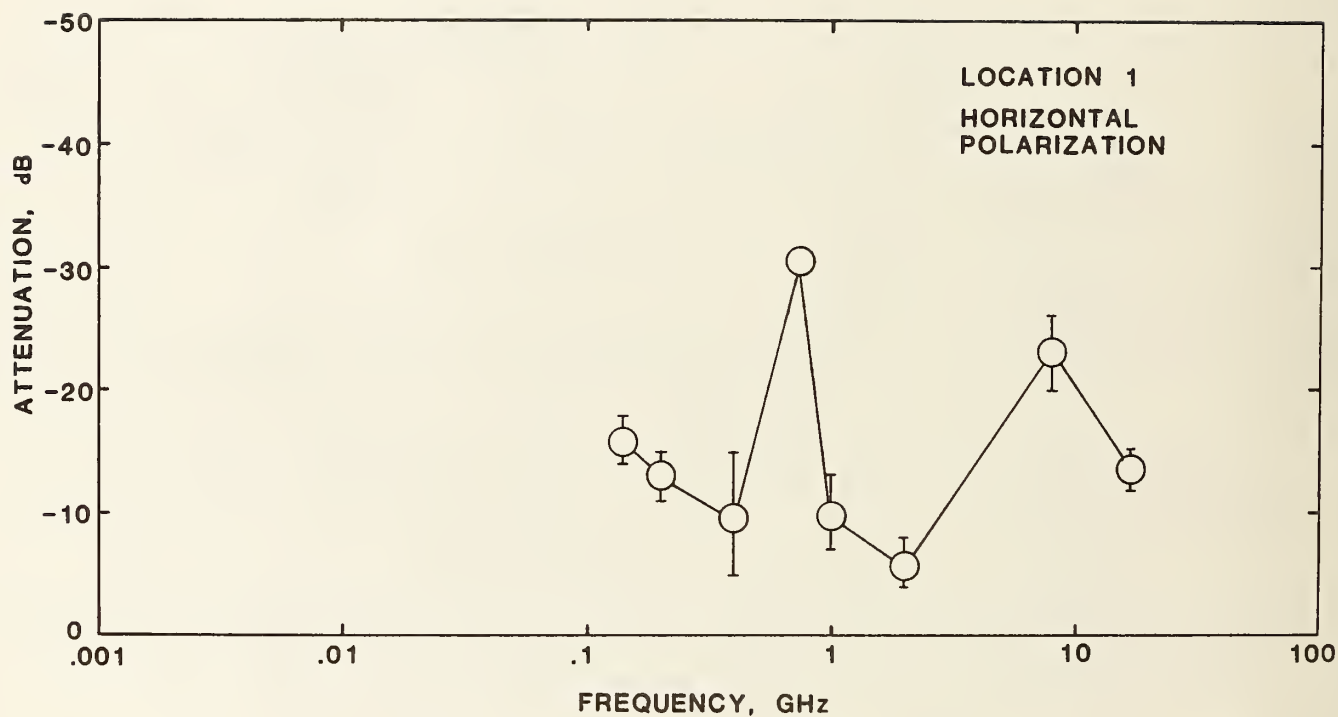


Figure 6.10. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 1 with horizontal polarization launched. The circles represent the average electric field attenuation, while the limit bars represent the highest and lowest attenuations observed at each frequency.

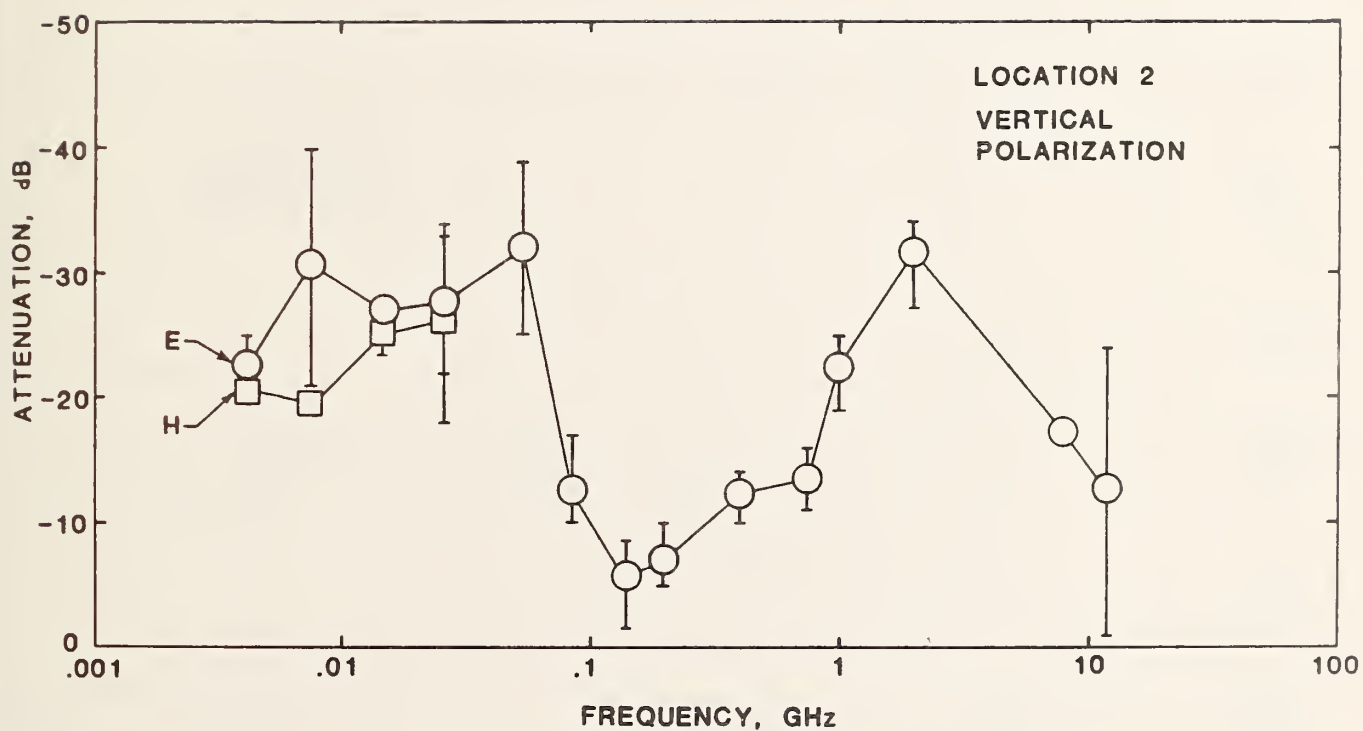


Figure 6.11. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 2 with vertical polarization launched. The circles represent average electric field attenuation, while the average magnetic field attenuation is shown as a square. The limit bars represent the highest and lowest attenuations observed at each frequency.

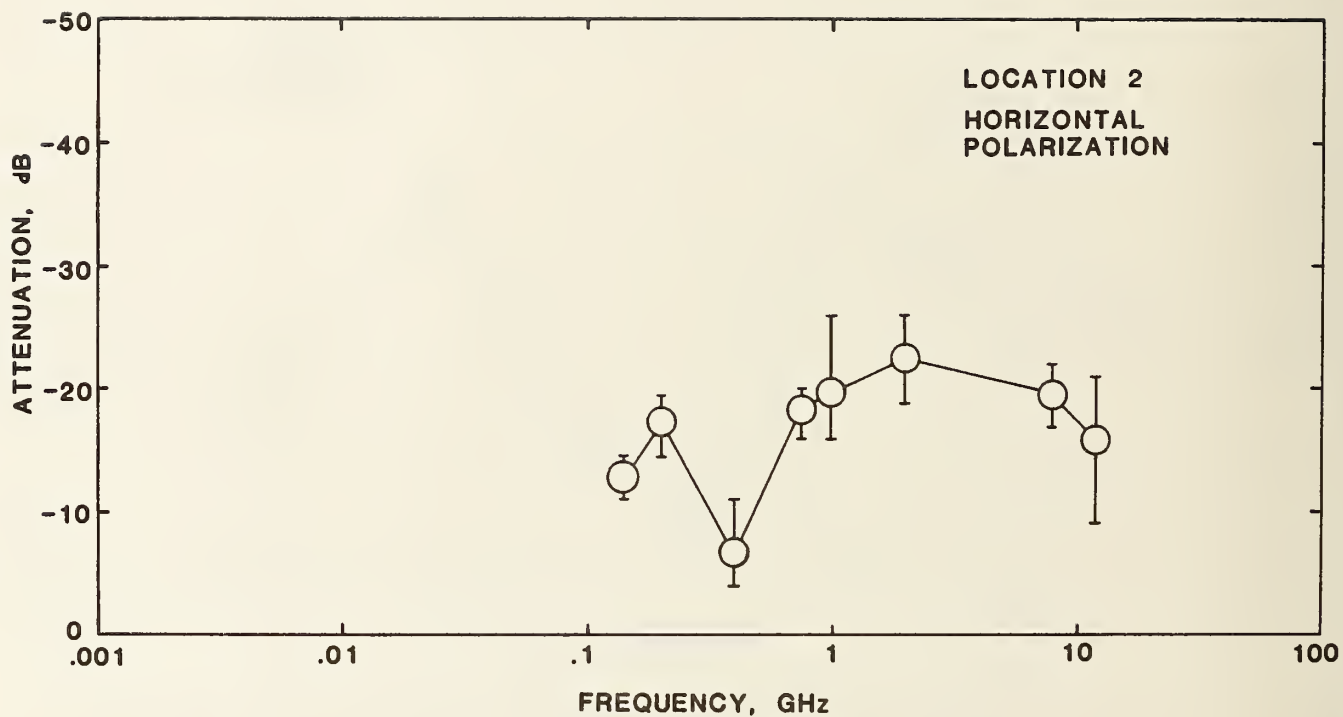


Figure 6.12. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 2 with horizontal polarization launched. The circles represent the average electric field attenuation, while the limit bars represent the highest and lowest attenuation observed at each frequency.

7. COMPUTER RESULTS AND CONCLUSIONS

7.1 Comparison With Experimental Results

To test the validity of the computer calculation of building attenuation, the program was run for a building that had been measured experimentally for electromagnetic shielding effectiveness. The structure chosen for comparison was the training module located at the Naval Training Equipment Center (NTEC) in Orlando, Florida. The experimental measurements of shielding effectiveness are already described in Section 6.2 of this report.

To input data into the computer program, the training module floor plan shown in figure 6.8, was redrawn, as shown in figure 7.1, to show the details of the walls, windows, and doors, based on observations made at the measurement site. Some assumptions and compromises were also made so that the building could be properly modelled by the computer. Looking at figure 6.8, the main questions were: 1) how to account for the rubber gaskets between the modules. The first question was solved by considering the open space between the modules as a fourth room, or room 4 as shown on figure 7.1. The walls were taken as MØ1 ("NULL MATERIAL") and windows of material MØ1 with a metal frame (M12 (STEEL)) were added on the three ends of the room. The "T" shape of ROOM Ø4 was created by using two rectangular shaped volumes and giving them the same name. For example, look at the last two data entry lines of the first table in figure 7.2(a) and notice that room 4 has two ceiling sections, one with dimensions 6.1Ø x 1.22 m and the second with 6.1Ø x Ø.15 m. The three windows, DA, DB and DB, shown in the figure were added so that the input resonance condition described in Section 3.5 would be taken into account.

The air conditioners (question 2) were considered closed doors in the model ("DC" on the diagram). The rubber gaskets (question 3) were modelled by ignoring them. They should have no shielding effectiveness for electromagnetic radiation.

When the experimental measurements were made on the shelter, some of the exterior doors were left open to provide ventilation for the equipment from the hot and humid conditions. Those doors, marked "DO" in figure 7.1, are modelled as "open" for the computer program so that the calculations can be properly compared with the experimental results. (Experimentally it was found that opening the doors had less than a 2 dB effect on the measurements. Since equipment failure was experienced with the doors closed, and since the experimental uncertainty was typically greater than 2 dB, the engineers in charge of the measurement made the decision to leave those doors open.)

The computer print-out for the calculation is shown in figure 7.2 (a-d). The first three tables list the wall data file B2Ø4W, the hole data file B2Ø4H, and hole types data file B2Ø4T. The next thirteen tables list the attenuation of each room (1-4) for each direction of input (1-5) for frequencies in the range of 1.Ø MHz to 1Ø GHz. In figures 7.3-7.6, the experimental data (open circles and squares)

is compared to the calculated data (closed triangles) where transmitter location 1 corresponds to computer direction D1, and transmitter location 2 corresponds to computer direction D2. Since experimental measurements were made only for room 1, the calculated data is displayed for just room 1.

By symmetry, a field projected from direction D3 should yield identical results with a field projected from D1. This is evident in the attenuation tables of figure 7.2 (a-d) where the column corresponding to D3 is identical with the results shown in the first column, D1. The fourth column, D4, is -60 dB at all frequencies and for all rooms. This is caused by the fact that the wall facing direction D4 has no openings which will allow penetration. Since that wall "shadows" all the other rooms, and since the computer model does not include external diffraction around corners, all of the rooms will have a -60 dB attenuation factor for direction D4. In figure 7.7, the room and door resonances of the test structure are shown. In the 0.1 GHz range the room resonance and door resonance for door "D0" dominate and drop the attenuation factor to zero. (Where the computer model calculated gain for a room, the attenuation factor was taken to be zero.) This effect was equally strong for both directions D1 and D2 and can be seen in figures 7.3 and 7.4. At around 1.0 GHz, the dominant resonant effect is due to door "DB". Since this door is only illuminated from direction D1, the calculated attenuation at 1.0 GHz is reduced for direction D1 (fig. 7.3) but not for direction D2 (fig. 7.4). This is in good agreement with the corresponding experimental measurements for those directions.

Overall, the fit between the calculated and experimental results are in good agreement. Given just the calculated data, it would be possible to estimate the shielding effectiveness of the training modules.

7.2 Recommendation for Further Work

One area in the model that could use further development is in the resonance calculations. The present approach essentially turns the resonances "on" or "off" and does not use any sophisticated techniques to properly weight the resonant effects. It should be possible to incorporate more advanced resonant models into the program. That task should be addressed in future work.

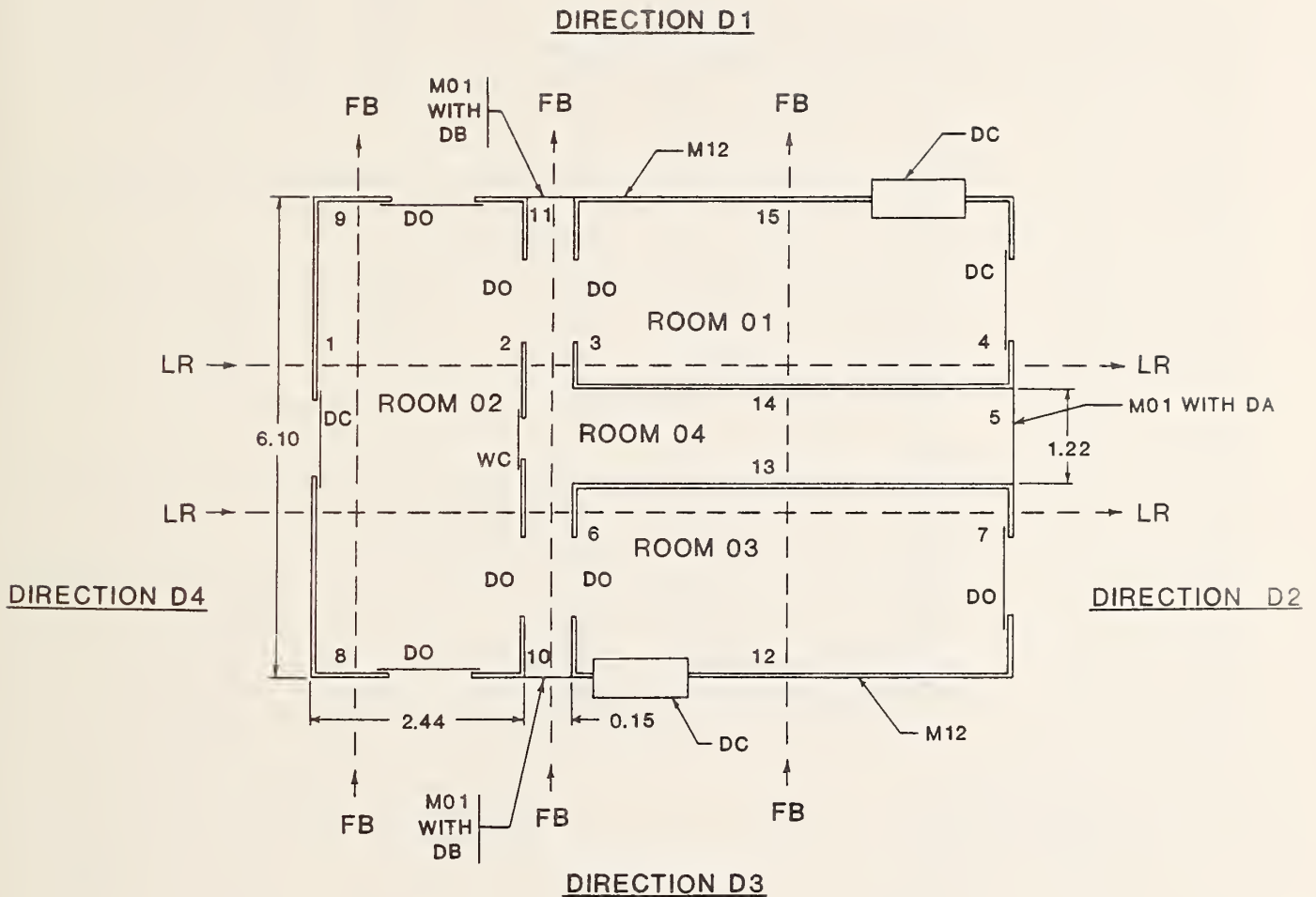


Figure 7.1. Building layout of training modules used for comparison of calculated versus experimental data. Notice that the air gap between the modules is considered "Room 04" for the computer model.

ENTER BUILDING IDENTIFICATION (E.G. '101')
 (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
 ? '204'
 BUILDING IDENTIFICATION ENTERED AS '204'
 ENTER NUMBER OF ROOMS IN BUILDING
 ? 4

WALL IDENTIFICATION			WALL PARAMETERS			
DIR	FROM	TO	MATERIAL	HEIGHT	WIDTH	THICKNESS
LR	D4	02	M12	2.44	6.10	.20
LR	02	04	M12	2.44	6.10	.20
LR	04	01	M12	2.44	2.44	.20
LR	01	D2	M12	2.44	2.44	.20
LR	04	D2	M01	2.44	1.22	.20
LR	04	03	M12	2.44	2.44	.20
LR	03	D2	M12	2.44	2.44	.20
FB	D3	02	M12	2.44	2.44	.20
FB	02	D1	M12	2.44	2.44	.20
FB	D3	04	M01	2.44	.15	.20
FB	04	D1	M01	2.44	.15	.20
FB	D3	03	M12	2.44	6.10	.20
FB	03	04	M12	2.44	6.10	.20
FB	04	01	M12	2.44	6.10	.20
FB	01	D1	M12	2.44	6.10	.20
UD	D5	01	M12	6.10	2.44	.20
UD	D5	02	M12	6.10	2.44	.20
UD	D5	03	M12	6.10	2.44	.20
UD	D5	04	M01	6.10	1.22	1.00
UD	01	D6	M01	6.10	.15	1.00
UD	01	D6	M12	6.10	2.44	.20
UD	02	D6	M12	6.10	2.44	.20
UD	03	D6	M12	6.10	2.44	.20
UD	04	D6	M01	6.10	1.22	1.00
UD	04	D6	M01	6.10	.15	1.00

DOOR AND WINDOW LOCATIONS

WALL IDENTIFICATION

ID	DIRECTION	FROM	TO
DC	LR	D4	02
DO	LR	02	04
DO	LR	04	01
DC	LR	01	D2
WC	LR	02	04
DA	LR	04	D2
DO	LR	02	04
DO	LR	04	03
DO	LR	03	D2
DC	FB	D3	03
DO	FB	D3	02
DO	FB	02	D1
DE	FB	D3	04
DB	FB	04	D1
DC	FB	01	D1

Figure 7.2(a). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

DOOR AND WINDOW PARAMETERS

```

*****
ID  MATERIAL FRAME  HEIGHT WIDTH  LAYER    DISTANCE
      MATERIAL          THICKNESS ABOVE FLR
=====
DA  M01      M12    2.44  1.22    .20    0.00
DB  M01      M12    2.44  .15     .20    0.00
DC  M12      M12    1.76  1.22    .20    0.00
DO  M01      M12    1.76  1.22    .20    0.00
WC  M12      M12    .61   .46    .20    0.00
WA  M10      M12    1.76  1.22    .20    0.00
=====

```

ATTENUATION AT A FREQUENCY OF 1.000E+06 HZ

```

*****
*          DIRECTIONS          *
* ROOMS *      1      2      3      4      5      *
*****
* 1 *      -32.42      -23.59      -32.42      -60.00      -28.73 *
* 2 *      -6.96      -23.31      -6.96      -60.00      -28.45 *
* 3 *      -35.71      -7.48      -35.71      -60.00      -32.03 *
* 4 *      -12.73      -3.91      -12.73      -60.00      -9.05 *
*****

```

ATTENUATION AT A FREQUENCY OF 2.000E+06 HZ

```

*****
*          DIRECTIONS          *
* ROOMS *      1      2      3      4      5      *
*****
* 1 *      -32.42      -23.59      -32.42      -60.00      -28.73 *
* 2 *      -6.96      -23.31      -6.96      -60.00      -28.45 *
* 3 *      -35.71      -7.48      -35.71      -60.00      -32.03 *
* 4 *      -12.73      -3.91      -12.73      -60.00      -9.05 *
*****

```

ATTENUATION AT A FREQUENCY OF 5.000E+06 HZ

```

*****
*          DIRECTIONS          *
* ROOMS *      1      2      3      4      5      *
*****
* 1 *      -32.42      -23.59      -32.42      -60.00      -28.73 *
* 2 *      -6.96      -23.31      -6.96      -60.00      -28.45 *
* 3 *      -35.71      -7.48      -35.71      -60.00      -32.03 *
* 4 *      -12.73      -3.91      -12.73      -60.00      -9.05 *
*****

```

ATTENUATION AT A FREQUENCY OF 1.000E+07 HZ

```

*****
*          DIRECTIONS          *
* ROOMS *      1      2      3      4      5      *
*****
* 1 *      -32.42      -23.59      -32.42      -60.00      -28.73 *
* 2 *      -6.96      -23.31      -6.96      -60.00      -28.45 *
* 3 *      -35.71      -7.48      -35.71      -60.00      -32.03 *
* 4 *      -12.73      -3.91      -12.73      -60.00      -9.05 *
*****

```

Figure 7.2(b). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

ATTENUATION AT A FREQUENCY OF 2.000E+07 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5

* 1 *	-32.42	-23.59	-32.42	-60.00	-28.73
* 2 *	-6.96	-23.31	-6.96	-60.00	-28.45
* 3 *	-35.71	-7.48	-35.71	-60.00	-32.03
* 4 *	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 5.000E+07 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5

* 1 *	-32.42	-23.59	-32.42	-60.00	-28.73
* 2 *	-6.96	-23.31	-6.96	-60.00	-28.45
* 3 *	-35.71	-7.48	-35.71	-60.00	-32.03
* 4 *	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 1.000E+08 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5

* 1 *	5.65	10.41	5.65	-60.00	-15.70
* 2 *	9.40	7.51	9.40	-60.00	-18.60
* 3 *	2.08	12.18	2.08	-60.00	-19.27
* 4 *	6.84	11.60	6.84	-60.00	-14.51

ATTENUATION AT A FREQUENCY OF 2.000E+08 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5

* 1 *	5.65	10.41	5.65	-60.00	-15.70
* 2 *	9.40	7.51	9.40	-60.00	-18.60
* 3 *	2.08	12.18	2.08	-60.00	-19.27
* 4 *	6.84	11.60	6.84	-60.00	-14.51

ATTENUATION AT A FREQUENCY OF 5.000E+08 HZ

* * * * *					
* DIRECTIONS *					
* ROOMS *	1	2	3	4	5

* 1 *	-10.11	-21.00	-10.11	-60.00	-26.14
* 2 *	-5.49	-21.78	-5.49	-60.00	-26.92
* 3 *	-14.39	-7.39	-14.39	-60.00	-30.42
* 4 *	8.51	-2.37	8.51	-60.00	-7.51

Figure 7.2(c). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

```

ATTENUATION AT A FREQUENCY OF 1.000E+09 HZ
*****
*          *          DIRECTIONS          *
* ROOMS *      1          2          3          4          5      *
*****
* 1 *      -10.11      -21.00      -10.11      -60.00      -26.14 *
* 2 *      -5.49      -21.78      -5.49      -60.00      -26.92 *
* 3 *     -14.39      -7.39      -14.39      -60.00      -30.42 *
* 4 *       8.51      -2.37       8.51      -60.00      -7.51 *
*****

```

```

ATTENUATION AT A FREQUENCY OF 2.000E+09 HZ
*****
*          *          DIRECTIONS          *
* ROOMS *      1          2          3          4          5      *
*****
* 1 *      -10.11      -21.00      -10.11      -60.00      -26.14 *
* 2 *      -5.49      -21.78      -5.49      -60.00      -26.92 *
* 3 *     -14.39      -7.39      -14.39      -60.00      -30.42 *
* 4 *       8.51      -2.37       8.51      -60.00      -7.51 *
*****

```

```

ATTENUATION AT A FREQUENCY OF 5.000E+09 HZ
*****
*          *          DIRECTIONS          *
* ROOMS *      1          2          3          4          5      *
*****
* 1 *     -32.42      -23.59      -32.42      -60.00      -28.73 *
* 2 *      -6.96      -23.31      -6.96      -60.00      -28.45 *
* 3 *     -35.71      -7.48      -35.71      -60.00      -32.03 *
* 4 *     -12.73      -3.91      -12.73      -60.00      -9.05 *
*****

```

```

ATTENUATION AT A FREQUENCY OF 1.000E+10 HZ
*****
*          *          DIRECTIONS          *
* ROOMS *      1          2          3          4          5      *
*****
* 1 *     -32.42      -23.59      -32.42      -60.00      -28.73 *
* 2 *      -6.96      -23.31      -6.96      -60.00      -28.45 *
* 3 *     -35.71      -7.48      -35.71      -60.00      -32.03 *
* 4 *     -12.73      -3.91      -12.73      -60.00      -9.05 *
*****

```

Figure 7.2(d). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

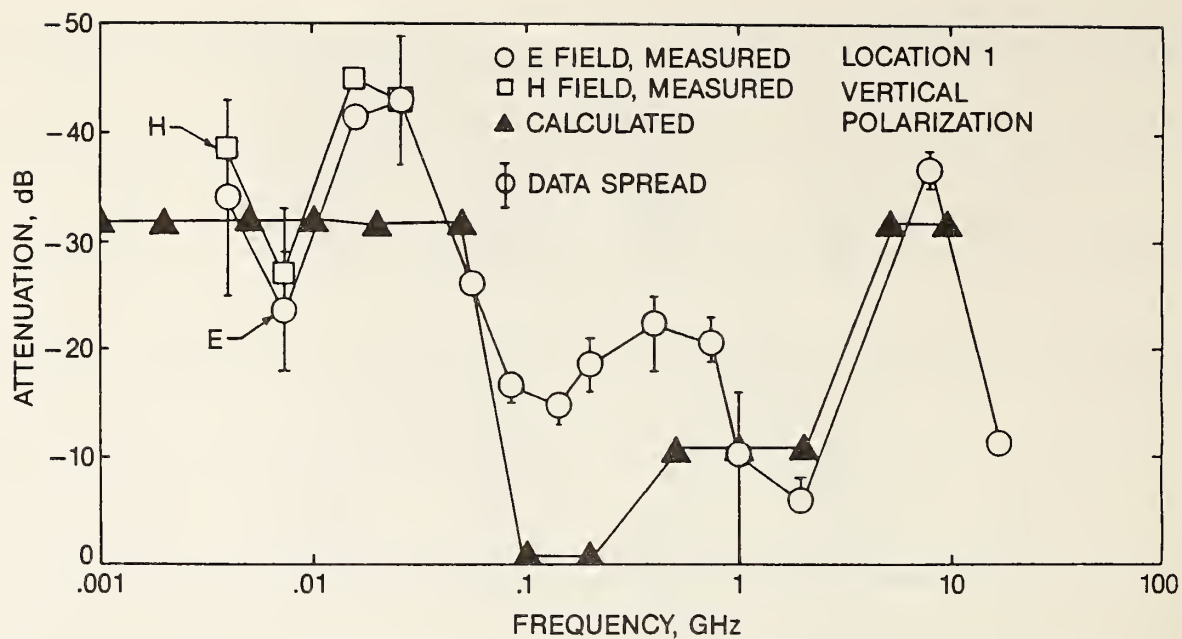


FIGURE 7.3 TRANSMITTER LOCATION 1, VERTICAL POLARIZATION DATA

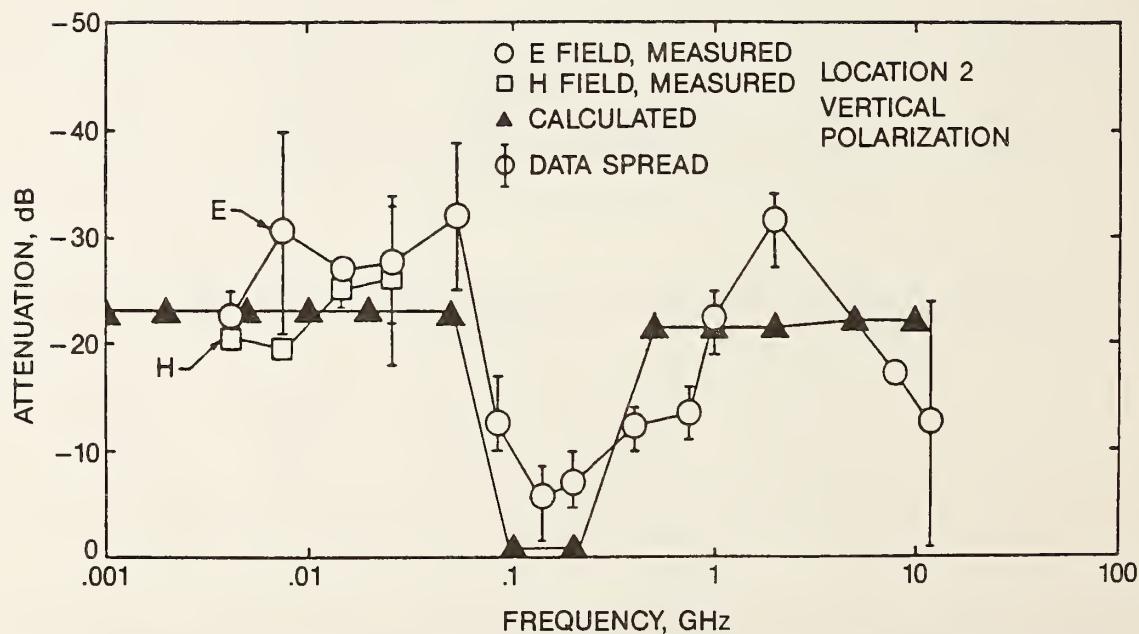


FIGURE 7.4 TRANSMITTER LOCATION 2, VERTICAL POLARIZATION DATA

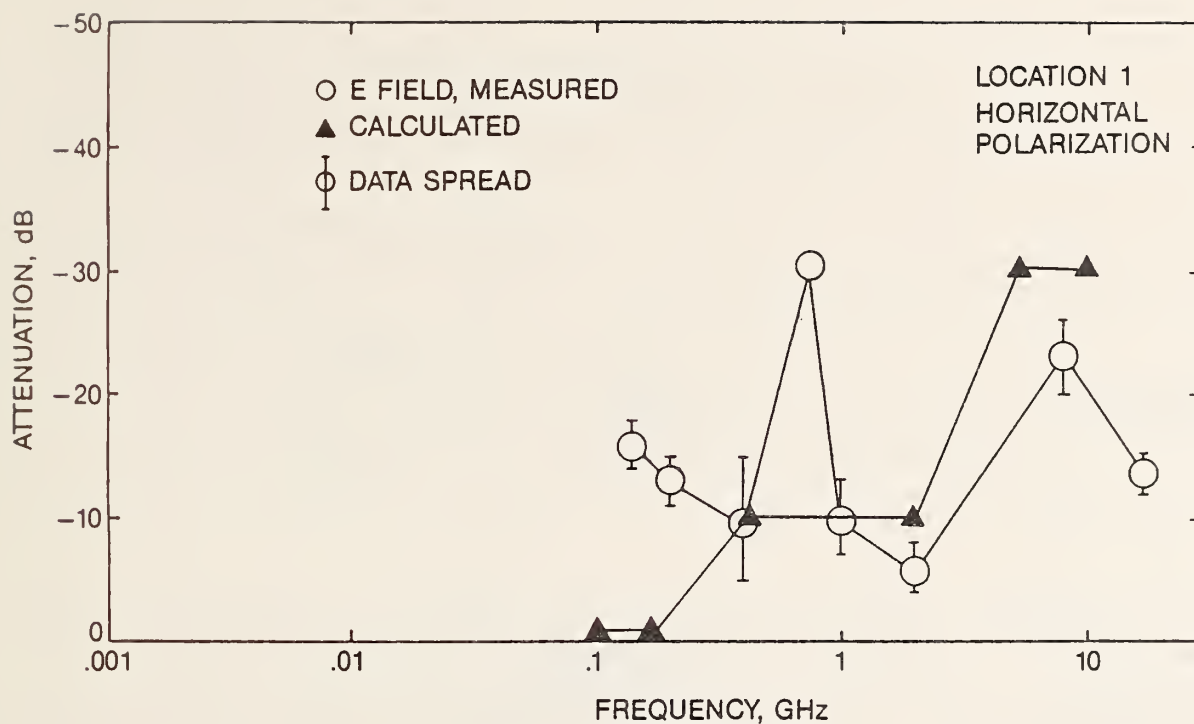


FIGURE 7.5 TRANSMITTER LOCATION 1, HORIZONTAL POLARIZATION DATA

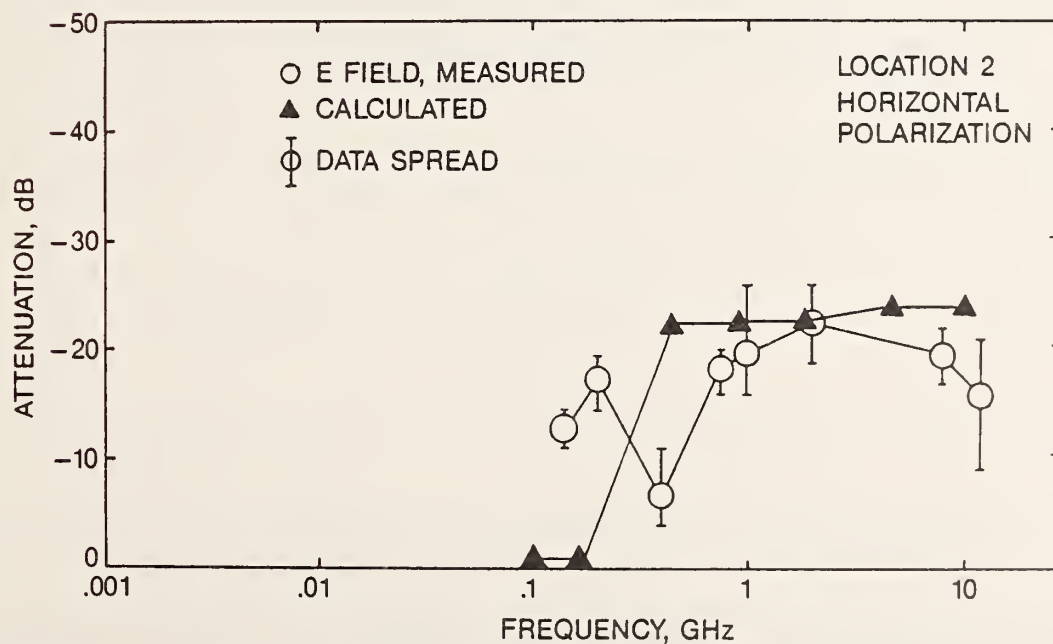


FIGURE 7.6 TRANSMITTER LOCATION 2, HORIZONTAL POLARIZATION DATA

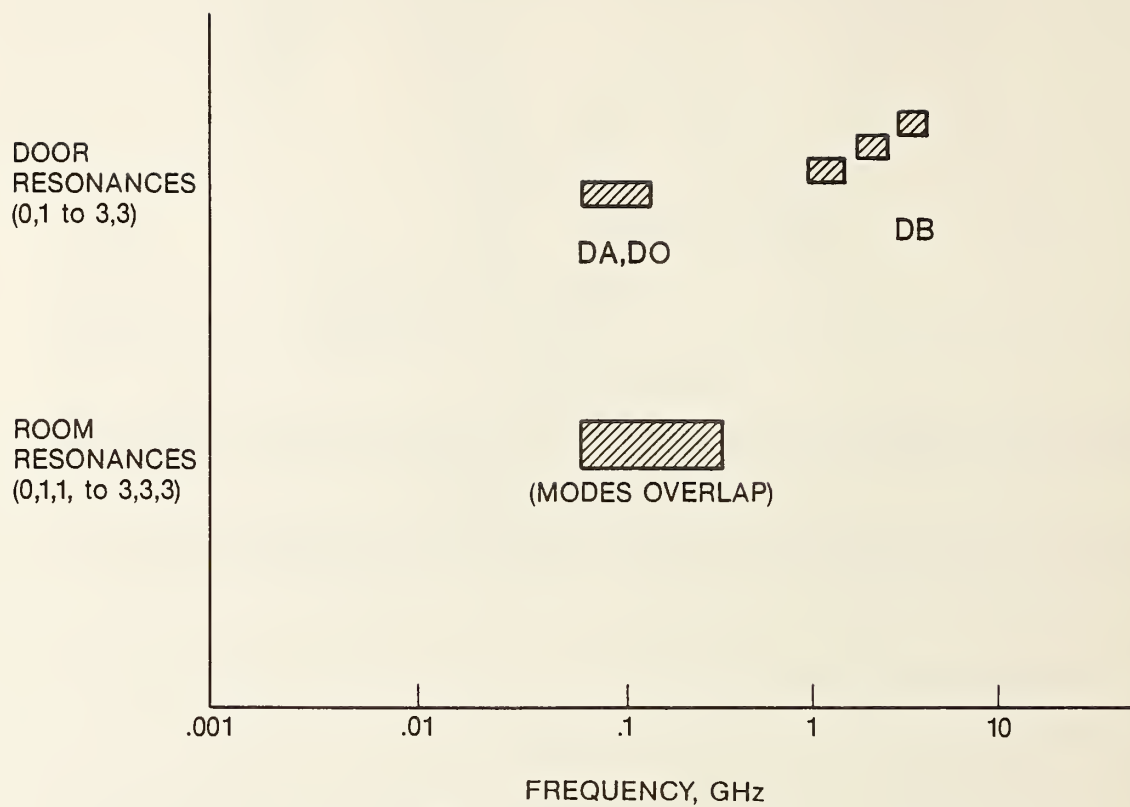


FIGURE 7.7 TEST STRUCTURE ROOM AND DOOR RESONANCES

8. REFERENCES

- [1] Cox, D. C.; Murray, R. R.; Norris, A. W. Measurements of 800 MHz radio transmission into buildings with metallic walls. To be published in Bell System Technical Journal.
- [2] Rice, L. P. Radio transmission into buildings on 35 and 150 MHz. Bell System Technical Journal 38: 197-210; 1959 January.
- [3] Smith, A. A., Jr. Attenuation of electric and magnetic fields by buildings. IEEE Trans. Electro-mag. Compat. EMC-20(3): 411-418; 1978 August.
- [4] Brennan, C. M.; Lambert, C. C.; Conner, C. G.; Roberts, G. F.; Flannery, W. T. EM shielding of building materials. Electro-Mechanics Co., Austin, Texas. Rome Air Development Center final report RADC-TR-67-446; 1968 February. 183 p. (AD-667-546).
- [5] Garrett, K. S.; Flannery, W. T.; Osburn, J. D.; Brennan, C. M.; Flory, R. K. A study of the shielding effectiveness of building materials. Electro-Mechanics Co., Austin, Texas. Rome Air Development Center final report RADC-TR-66-178; 1966 May. 382 p. (AD-485-027/7ST).
- [6] Johnk, Carl T.A. Engineering electromagnetic fields and waves. New York: John Wiley and Sons; 1975. 655 p.
- [7] Denny, H. W.; Holland, L. D.; Robinette, S. L.; Woody, J. A. Grounding, bonding, and shielding practices and procedures for electronic equipments and facilities. Vol. I: Fundamental considerations. Engineering Experiment Station, Georgia Institute of Technology. Federal Aviation Administration final report FAA-RD-75-215; 1975 December. 399 p. (AD-A022-332).
- [8] Campi, M. Survey and review of building shielding to electromagnetic waves from EMP. Harry Diamond Laboratories, Adelphi, Md. Final report HDL-TM-78-23; 1978 November. 109 p. (AD-B034-726L).
- [9] Jakubec, L. G.; Ohta, H. H. Proposed specifications for electromagnetic shielding of enclosures and buildings. Genistron, Incorporated, Los Angeles, CA. Phase I final report to U.S. Naval Civil Engineering Laboratory on contract NBy-32220; 1963 July. (AD-417-699).
- [10] Jarva, W. Shielding efficiency calculation methods for screening, waveguide ventilation panels, and other perforated electromagnetic shields. Proc. of the Seventh Conf. on Radio Interference Reduction and Electronic Compatibility. 1961 November; Armour Research Foundation of Ill. Inst. of Technology, and Professional Group on Radio Frequency Interference of the Inst. of Radio Engineers.
- [11] MacFarlane, G. G. Surface impedance of an infinite parallel-wire grid at oblique angles of incidence. Proc. IEE 93(IIIA): 1523-1527; 1946.
- [12] Hill, D. A.; Wait, J. R. Scattering of a transient plane wave by a periodic grating. Radio Sci. 11(6): 1003-1009; 1971 November.
- [13] Hill, D. A.; Wait, J. R. Electromagnetic scattering of an arbitrary plane wave by two nonintersecting perpendicular wire grids. Can. J. Phys. 52(3): 227-237; 1974.
- [14] Hill, D. A.; Wait, J. R. Electromagnetic scattering of an arbitrary plane wave by a wire mesh with bonded junctions. Can. J. Phys. 54(4): 353-361; 1976.
- [15] Kontorovich, M. I.; Petrun'kin, V. Yu.; Yesephkina, N. A.; Astrakhan, M. I. The coefficient of reflection of a plane electromagnetic wave from a plane wire mesh. Radio Eng. Elect. Phys. 7: 222-231; 1962.
- [16] Astrakhan, M. I. Reflecting and screening properties of plane wire grids. Radio Engineering. 23(1): 76-83; 1968.
- [17] James, W. L. Dielectric properties of wood and hardboard: variation with temperature, frequency, moisture content, and grain orientation. USDA Forest Service, Forest Products Laboratory, Madison, Wisc. USDA Forest Service Research Paper FPL 245; 1975. 32 p. (AD-A015-654).
- [18] Johnson, M. K. Dielectric constant and loss tangent determination for various dielectric materials to be used at the Trestle site. Braddock, Dunn, and McDonald, Inc., Albuquerque, New Mexico. Prepared for U.S. Air Force Trestle program. Report BDM/A-83-74-TR; 1974 July 4. 88 p. (AD-B008-055L).

APPENDIX 9.1

TEST PLAN

BUILDING ATTENUATION MEASUREMENTS

A.1 Purpose

This plan outlines a method for measuring the attenuation of buildings over all or part of the frequency range covered in this report: 10 kHz - 10 GHz. In order to compare measured attenuation data with computed data (generated by the computer program developed for this report), the building must be square or rectangular, or made up of adjoining squares or rectangles; and the test signals must be perpendicularly incident on the building walls.

A.2 Frequency Coverage

Apart from limitations imposed by the availability of portable sources and launching antennas, the choice of measurement frequencies may be determined by factors such as

- regions of low attenuation in the computed data.
- the frequencies of strong ambient signals.
- frequencies at which a given device is known to have highest susceptibility.

If there are not worrisome frequencies, the project engineer can make the choice of measurement frequencies over the 10 kHz - 10 GHz range.

Table A-1 lists the frequencies at which an NBS team made building attenuation measurements on an Army installation. Other columns in the table indicate distance increments between locations on the measurement grid, the type of field to be measured and the type to be launched, and the electric field polarization. A measurement grid is shown in figure A.1 for a single-room building or for a single room within a building, though the size, shape, and construction of a particular building may permit it to be characterized by many fewer measurement locations (see section A.3).

A.3 Number and location of measurements

A.3.1 For a free-standing, rectangular building with a single room, a field will be launched perpendicular to all four faces of the building in turn. Where many rooms exist, only some of which are of interest, or the general building geometry dictates, some of the four faces may not be used.

A.3.2 The exact pattern of measurement location is determined by the frequency and is adaptive. (Ref. figure A.1).

Table A.1. Tabulation of the frequencies and signal properties for a set of electromagnetic field attenuation measurements.

Frequency	Δ , Meters ^{1,2}	Type of Measurement	Field Launched	Polarization ³
180 kHz	8	E & H	Primarily H	Vert.
3.5 MHz	8	E & H	E X H	"
7 MHz	4	E	E X H	"
14 MHz	4	E	E X H	"
28 MHz	2	E	E X H	"
54 MHz	2	E	E X H	"
140 MHz	1*	E	E X H	"
200 MHz	1*	E	E X H	"
400 MHz	1*	E	E X H	"
750 MHz	1*	E	E X H	"
1000 MHz	1*	E	E X H	Circular or Horiz.
2000 MHz	1*	E	E X H	"
4000 MHz	1*	E	E X H	"
8000 MHz	1*	E	E X H	"
12000 MHz	1*	E	E X H	"
18000 MHz	1*	E	E X H	"

Notes

1. * Actual position will be varied
± 1/4 meter to achieve highest reading.
2. Δ is spacing of locations for sequence 3 and above.
3. Circular polarization will be used above 1 GHz if adequate signals are received inside building.
If signal levels are too low a high gain horizontally polarized launching antenna will be used.

A.3.2.1 First, the incident field is measured 5 meters from the face of the building at three locations as indicated.

A.3.2.2 Next, a 1 meter by 1 meter grid is established in the area to be measured. A line, two meters inside the front wall is measured every meter to within 2 meters of the side walls. This will be called sequence A.

A.3.2.3 The highest reading is noted and a line of points Δ meters apart (see Table A.1), perpendicular to the first line is measured, until a total-field measurement is obtained which is equal to or less than the lowest reading obtained in sequence A. This will be called sequence B. Note, however, if this line of measurements is within ± 3 meters of the center line of the building, this sequence may be eliminated.

A.3.2.4 A line of points Δ meters apart is now measured along the center line of the building to the center of the room. This is called sequence C.

A.3.2.5 If no other faces of the building are to be excited which are perpendicular to the first face, then a last sequence D across the middle of the room will be taken at spacings of Δ meters.

A.3.2.6 If L/Δ or $W/\Delta \leq 5$, (where L & W are dimensions of the room being measured) then at least 5 measurement points will be taken in sequences B, C, and D. The nearest full meter value for Δ will be chosen which will result in at least 5 measurement points within the room's dimensions.

A.3.2.7 All readings will be taken at a height of 1.5 meters except those at 140 MHz when a second set will be taken at a height of 1 meter.

A.3.2.8 Data will be recorded for the X, Y, and Z orthogonal components, plus the equivalent vector sum.

A.3.3 In buildings with a large door opening, data will also be taken with the door open at the frequency where the vertical dimension of the opening is $\lambda/2$, and at 7 MHz, 400 MHz, and 4 GHz.

Again, we emphasize that the test plan may be modified for each test site by the engineers in charge to conform to any constraints imposed by time or the physical layout of the structure. Although the test plan recommends that measurements be grouped in four locations (fig. A.1), the engineers in charge may decide that three (or in some cases, two) groups are sufficient to determine the shielding effectiveness of the buildings.

A.4 Data Presentation

Attenuation measurements can be summarized in tables and also plotted. The table for each building, room, and transmitter location can list the frequencies of the test, the mean values of attenuation measured, and uncertainties of the measurement (one standard deviation) shown in parentheses. The mean value of attenuation is determined by averaging all grid point measurements in a particular building or room, for a particular frequency for each transmitter location:

$$\bar{X} = \frac{1}{N} \sum x_i$$

where \bar{X} is the sample mean, N is the number of grid points measured, and x_i represents the individual attenuations at each measured grid point. The standard deviation is defined as the square root of the sample variance:

$$\sigma = \sqrt{s^2}$$

where

$$s^2 = \frac{1}{N-1} \sum (x_i - \bar{X})^2$$

and where σ is the standard deviation, s^2 is the sample variance, and x_i and \bar{X} are already defined. The graphs show the mean attenuation at each measurement frequency, with the one-standard-deviation limits as error bars.

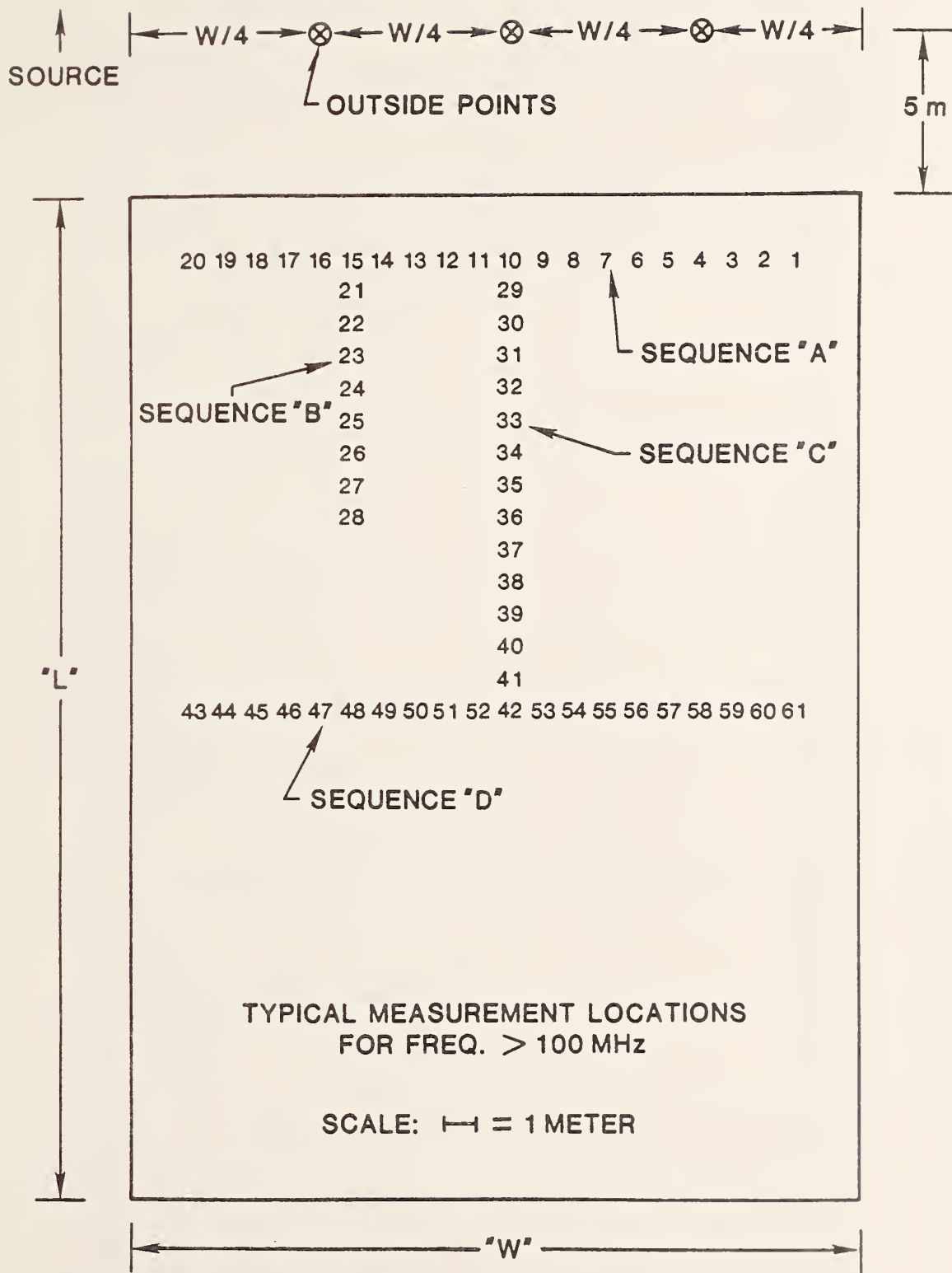


Figure A.1 Typical measurement locations for frequencies greater than 100 MHz

Appendix 9.2 Listing of Computer Program SMATDB

1	PROGRAM SMATDB	SMATDB	1
2	C** INPUT MATERIAL ATTENUATION AND REFLECTION COEFFICIENT INTO	SMATDB	2
3	C** ARRAYS AND THEN STORE DATA IN A PERMANENT FILE*	SMATDB	3
4	**	SMATDB	4
5	*****	COMM	1
6	*** COMMON FOR DATABASE OF MATERIAL PROPERTIES	***COMM	2
7	*****	COMM	3
8	INTEGER MMAX	COMM	4
9	PARAMETER (MMAX=100)	COMM	5
10	COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX),	COMM	6
11	\$ MFREQ(MMAX,7), MERR, MTOT	COMM	7
12	COMMON /MATC/MAT(MMAX),MATDESC(MMAX)	COMM	8
13	INTEGER MTOT, MERR	COMM	9
14	REAL MATTEN, MRCOEF, MFREQ, QA, QR	COMM	10
15	CHARACTER * 3 MAT	COMM	11
16	CHARACTER * 70 MATDESC	COMM	12
17	*****	COMM	13
18	*****	COMM	14
19	INTEGER R,C, COMMAND	SMATDB	6
20	CHARACTER * 3 MATID	SMATDB	7
21	C* INITIALIZE ARRAYS	SMATDB	8
22	DATA MAT / 100 * ' ' /	SMATDB	9
23	DATA MATDESC / 100 * ' ' /	SMATDB	10
24	DATA MFREQ / 700 * 0.0 /	SMATDB	11
25	DATA MATTEN / 700 * 0.0 /	SMATDB	12
26	DATA QA / 100 * 0.0 /	SMATDB	13
27	DATA MRCOEF / 700 * 0.0 /	SMATDB	14
28	DATA QR / 100 * 0.0 /	SMATDB	15
29	C* ENTER COMMANDS	SMATDB	16
30	PRINT *, '(1) CREATE NEW DATABASE (2) ADD TO EXISTING DATA ',	SMATDB	17
31	Z 'BASE'	SMATDB	18
32	READ *, COMMAND	SMATDB	19
33	IF (COMMAND .EQ. 2) THEN	SMATDB	20
34	CALL LMATTER	SMATDB	21
35	IF (MERR .NE. 0) CALL ERROR(5)	SMATDB	22
36	ENDIF	SMATDB	23
37	10 PRINT*, '(1) NEXT DATA ENTRY (2) CHANGE (3) DISPLAY (4) ABORT',	SMATDB	24
38	\$ ' (5) QUIT'	SMATDB	25
39	READ *, COMMAND	SMATDB	26
40	PRINT *	SMATDB	27
41	IF (COMMAND .EQ. 1) THEN	SMATDB	28
42	CALL NEXT	SMATDB	29
43	ELSE IF (COMMAND .EQ. 2) THEN	SMATDB	30
44	CALL CHANGE	SMATDB	31
45	ELSE IF (COMMAND .EQ. 3) THEN	SMATDB	32
46	CALL DISPLAY	SMATDB	33
47	ELSE IF (COMMAND .EQ. 4) THEN	SMATDB	34
48	PRINT*	SMATDB	35
49	PRINT*, 'PROGRAM ABORTED AT YOUR REQUEST'	SMATDB	36
50	PRINT*	SMATDB	37
51	STOP	SMATDB	38
52	ELSE IF (COMMAND .EQ. 5) THEN	SMATDB	39
53	CALL QUIT	SMATDB	40
54	STOP	SMATDB	41
55	ENDIF	SMATDB	42
56	GOTO 10	SMATDB	43
57	END	SMATDB	44

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	NONE		UNUSED/*S*	INTEGER	
COMMAND	157B			INTEGER	
MAT	0B	/MATC/		CHAR*3	100
MATDESC	36B	/MATC/		CHAR*70	100
MATID	NONE		UNUSED/*S*	CHAR*3	
MATTEN	0B	/MATN/		REAL	700
MERR	4374B	/MATN/		INTEGER	
MFREQ	3100B	/MATN/		REAL	700
MRCOEF	1274B	/MATN/		REAL	700
MTOT	4375B	/MATN/		INTEGER	
QA	2570B	/MATN/		REAL	100
QR	2734B	/MATN/		REAL	100
R	NONE		UNUSED/*S*	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

MMAX	INTEGER	100
------	---------	-----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS----- -NAME-----TYPE-----ARGS-----CLASS-----

CHANGE		0	SUBROUTINE	LMATTER		0	SUBROUTINE
DISPLAY		0	SUBROUTINE	NEXT		0	SUBROUTINE
ERROR		1	SUBROUTINE	QUIT		0	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	24B	37
----	-----	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

SMATDB	5B	0
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	160B = 112
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.081 SECONDS

1		SMATDB	45
2	SUBROUTINE NEXT	SMATDB	46
3	*****	COMM	1
4	*** COMMON FOR DATABASE OF MATERIAL PROPERTIES	*** COMM	2
5	*****	COMM	3
6	INTEGER MMAX	COMM	4
7	PARAMETER (MMAX=100)	COMM	5
8	COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX),	COMM	6
9	\$ MFREQ(MMAX,7), MERR, MTOT	COMM	7
10	COMMON /MATC/MAT(MMAX),MATDESC(MMAX)	COMM	8
11	INTEGER MTOT, MERR	COMM	9
12	REAL MATTEN, MRCOEF, MFREQ, QA, QR	COMM	10
13	CHARACTER * 3 MAT	COMM	11
14	CHARACTER * 70 MATDESC	COMM	12
15	*****	COMM	13
16	*****	COMM	14
17	INTEGER R,C, VAL, INDEX	SMATDB	48
18	CHARACTER * 3 MATID	SMATDB	49
19	LOGICAL ENTERED	SMATDB	50
20 10	PRINT *, 'MATERIAL I.D.? (E.G. 'M05' OR 'M12')'	SMATDB	51
21	READ *, MATID	SMATDB	52
22	IF (MATID(1:1) .NE. 'M') THEN	SMATDB	53
23	PRINT *, 'FIRST CHARACTER MUST BE AN M. TRY AGAIN'	SMATDB	54
24	GOTO 10	SMATDB	55
25	ENDIF	SMATDB	56
26	C* CHECK IF THIS MATERIAL IS ALREADY ENTERED	SMATDB	57
27	ENTERED = .FALSE.	SMATDB	58
28	DO 20 R=1,MMAX	SMATDB	59
29	IF (MAT(R) .EQ. MATID) THEN	SMATDB	60
30	ENTERED = .TRUE.	SMATDB	61
31	INDEX = R	SMATDB	62
32	ENDIF	SMATDB	63
33 20	CONTINUE	SMATDB	64
34	C* ENTER NEW DATA IF MATERIAL NOT ALREADY ENTERED	SMATDB	65
35	IF (ENTERED) THEN	SMATDB	66
36	PRINT *, 'MATERIAL ',MATID,' ALREADY ENTERED'	SMATDB	67
37	ELSE	SMATDB	68
38	INDEX = VAL (MATID(2:3))	SMATDB	69
39	PRINT *, 'INDEX:',INDEX	SMATDB	70
40	MAT (INDEX) = MATID	SMATDB	71
41	PRINT *, 'ENTER ONE LINE DESCRIPTION OF MATERIAL'	SMATDB	72
42	READ *, MATDESC (INDEX)	SMATDB	73
43	PRINT *, 'ENTER 7 ATTENUATION VALUES FROM LOW TO HIGH FREQ'	SMATDB	74
44	READ *, (MATTEN(INDEX,C), C=1,7)	SMATDB	75
45	PRINT *, 'ENTER ATTENUATION QUALITY PERCENT'	SMATDB	76
46	READ *, QA (INDEX)	SMATDB	77
47	PRINT *, 'ENTER 7 REFLECTION COEFFS FROM LOW TO HIGH FREQ'	SMATDB	78
48	READ *, (MRCOEF(INDEX,C), C=1,7)	SMATDB	79
49	PRINT *, 'ENTER REFLECTION COEFFICIENT QUALITY PERCENT'	SMATDB	80
50	READ *, QR (INDEX)	SMATDB	81
51	PRINT *	SMATDB	82
52	ENDIF	SMATDB	83
53	RETURN	SMATDB	84
54	END	SMATDB	85

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	357B		INTEGER		
ENTERED	362B		LOGICAL		
INDEX	360B		INTEGER		
MAT	0B	/MATC/	CHAR*3	100	
MATDESC	36B	/MATC/	CHAR*70	100	

FTN 5.1+552 83/12/24. 09.48.06 PAGE 4
SUBROUTINE NEXT 74/175 OPT=0

MATID	361B		CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOEF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	356B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

MMAX	INTEGER	100
------	---------	-----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

VAL	INTEGER	1	FUNCTION
-----	---------	---	----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	5B	20
20	INACTIVE	DO-TERM 33

--ENTRY POINTS--(LO=A)

-NAME-----ADDRESS--ARGS---

NEXT	4B	0
------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	366B = 246
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.117 SECONDS

1		SMATDB	86
2		SMATDB	87
3	SUBROUTINE DISPLAY	SMATDB	88
4	*****	COMM	1
5	*** COMMON FOR DATABASE OF MATERIAL PROPERTIES	***COMM	2
6	*****	COMM	3
7	INTEGER MMAX	COMM	4
8	PARAMETER (MMAX=100)	COMM	5
9	COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX),	COMM	6
10	\$ MFREQ(MMAX,7), MERR, MTOT	COMM	7
11	COMMON /MATC/MAT(MMAX),MATDESC(MMAX)	COMM	8
12	INTEGER MTOT, MERR	COMM	9
13	REAL MATTEN, MRCOEF, MFREQ, QA, QR	COMM	10
14	CHARACTER * 3 MAT	COMM	11
15	CHARACTER * 70 MATDESC	COMM	12
16	*****	COMM	13
17	*****	COMM	14
18	INTEGER R,C, COMMAND	SMATDB	90
19	CHARACTER * 3 MATID	SMATDB	91
20	LOGICAL FOUND	SMATDB	92
21	PRINT *, '(1) ALL MATERIALS OR (2) ONE MATERIAL'	SMATDB	93
22	READ *, COMMAND	SMATDB	94
23	PRINT *	SMATDB	95
24	IF (COMMAND .EQ. 1) THEN	SMATDB	96
25	PRINT *	SMATDB	97
26	DO 10 R = 1,MMAX	SMATDB	98
27	MATID = MAT(R)	SMATDB	99
28	IF (MATID(1:1).EQ.'M') THEN	SMATDB	100
29	PRINT * , MAT(R)	SMATDB	101
30	PRINT * , MATDESC (R)	SMATDB	102
31	PRINT *, 'FREQUENCY: ' ,(MFREQ(R,C),C=1,7)	SMATDB	103
32	PRINT * , 'ATTENUATION: ' ,(MATTEN(R,C), C=1,7)	SMATDB	104
33	PRINT *, 'ATTENUATION QUALITY PERCENT: ',QA(R)	SMATDB	105
34	PRINT * , 'REFLECTION: ' , (MRCOEF(R,C), C=1,7)	SMATDB	106
35	PRINT *, 'REFLECTION COEF QUALITY PERCENT: ',QR(R)	SMATDB	107
36	PRINT *	SMATDB	108
37	ENDIF	SMATDB	109
38 10	CONTINUE	SMATDB	110
39	ELSEIF (COMMAND .EQ. 2) THEN	SMATDB	111
40	PRINT *, 'SPECIFY ID OF MATERIAL TO BE PRINTED (E.Q. M05)'	SMATDB	112
41	READ *, MATID	SMATDB	113
42	FOUND = .FALSE.	SMATDB	114
43	DO 20 R = 1,MMAX	SMATDB	115
44	IF (MAT(R) .EQ. MATID) THEN	SMATDB	116
45	PRINT * , MAT(R)	SMATDB	117
46	PRINT * , MATDESC(R)	SMATDB	118
47	PRINT *, 'FREQUENCY: ' ,(MFREQ(R,C),C=1,7)	SMATDB	119
48	PRINT * , 'ATTENUATION: ' ,(MATTEN(R,C), C=1,7)	SMATDB	120
49	PRINT *, 'ATTENUATION QUALITY PERCENT: ',QA(R)	SMATDB	121
50	PRINT * , 'REFLECTION: ' ,(MRCOEF(R,C), C=1,7)	SMATDB	122
51	PRINT *, 'REFLECTION QUALITY PERCENT: ',QR(R)	SMATDB	123
52	FOUND = .TRUE.	SMATDB	124
53	ENDIF	SMATDB	125
54 20	CONTINUE	SMATDB	126
55	ENDIF	SMATDB	127
56	IF (.NOT. FOUND) THEN	SMATDB	128
57	PRINT * , 'MATERIAL ' , MATID, ' NOT FOUND'	SMATDB	129
58	ENDIF	SMATDB	130
59	PRINT *	SMATDB	131
60	RETURN	SMATDB	132
61	END	SMATDB	133

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	511B		INTEGER	
COMMAND	512B		INTEGER	
FOUND	514B		LOGICAL	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	513B		CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOEF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	510B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

MMAX	INTEGER	100
------	---------	-----

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

10	INACTIVE	DO-TERM	38
20	INACTIVE	DO-TERM	54

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

DISPLAY	4B	0
---------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	525B = 341
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.172 SECONDS

1		SMATDB	134
2		SMATDB	135
3	INTEGER FUNCTION VAL(String)	SMATDB	136
4	C** RETURNS THE INTEGER VALUE OF A STRING.	SMATDB	137
5	INTEGER NUMBER, X, L, EXP, DIGIT, GETLEN	SMATDB	138
6	CHARACTER * (*) STRING	SMATDB	139
7	L = GETLEN(String)	SMATDB	140
8	NUMBER = 0	SMATDB	141
9	DO 10 X = L, 1, -1	SMATDB	142
10	EXP = L - X	SMATDB	143
11	DIGIT = ICHAR(String(X:X)) - 16	SMATDB	144
12	NUMBER = NUMBER + DIGIT*10**EXP	SMATDB	145
13 10	CONTINUE	SMATDB	146
14	VAL = NUMBER	SMATDB	147
15	RETURN	SMATDB	148
16	END	SMATDB	149

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

DIGIT	76B		INTEGER
EXP	75B		INTEGER
L	74B		INTEGER
NUMBER	72B		INTEGER
STRING	1	DUMMY-ARG	CHAR*(*)
VAL	71B		INTEGER
X	73B		INTEGER

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

10	INACTIVE	DO-TERM	13
----	----------	---------	----

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS---ARGS---

VAL	6B	1
-----	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.041 SECONDS

1		SMATDB	150
2		SMATDB	151
3	INTEGER FUNCTION GETLEN (STRING)	SMATDB	152
4	C	SMATDB	153
5	C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING	SMATDB	154
6	C	SMATDB	155
7	C	SMATDB	156
8	C ARGUMENT DEFINITIONS --	SMATDB	157
9	C INPUT ARGUMENTS	SMATDB	158
10	C STRING - STRING WHOSE LENGTH IS TO BE DETERMINED	SMATDB	159
11	C	SMATDB	160
12	C CHARACTER * (*) STRING	SMATDB	161
13	C	SMATDB	162
14	C FUNCTION PARAMETERS	SMATDB	163
15	C CHARACTER * 1 BLANK	SMATDB	164
16	C PARAMETER (BLANK = ' ')	SMATDB	165
17	C	SMATDB	166
18	C LOCAL VARIABLES	SMATDB	167
19	C INTEGER NEXT	SMATDB	168
20	C	SMATDB	169
21	C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK	SMATDB	170
22	C DO 10 NEXT = LEN(STRING),1,-1	SMATDB	171
23	C IF (STRING(NEXT : NEXT) .NE. BLANK) THEN	SMATDB	172
24	C GETLEN = NEXT	SMATDB	173
25	C RETURN	SMATDB	174
26	C ENDF	SMATDB	175
27	C 10 CONTINUE	SMATDB	176
28	C	SMATDB	177
29	C ALL CHARACTERS ARE BLANKS	SMATDB	178
30	C GETLEN = 0	SMATDB	179
31	C	SMATDB	180
32	C RETURN	SMATDB	181
33	C END	SMATDB	182

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

GETLEN	63B	INTEGER
NEXT	64B	INTEGER
STRING	1 DUMMY-ARG	CHAR*(*)

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

BLANK CHAR*1

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

LEN	INTEGER	1	INTRINSIC
-----	---------	---	-----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	27
----	----------	---------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

GETLEN	6B	1
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	70B = 56
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.038 SECONDS

1		SMATDB	183
2		SMATDB	184
3	SUBROUTINE QUIT	SMATDB	185
4	*****	COMM	1
5	*** COMMON FOR DATABASE OF MATERIAL PROPERTIES ***	COMM	2
6	*****	COMM	3
7	INTEGER MMAX	COMM	4
8	PARAMETER (MMAX=100)	COMM	5
9	COMMON /MATN/ MATTEN(MMAX,7), MRCOE(MMAX,7), QA(MMAX), QR(MMAX),	COMM	6
10	\$ MFREQ(MMAX,7), MERR, MTOT	COMM	7
11	COMMON /MATC/ MAT(MMAX), MATDESC(MMAX)	COMM	8
12	INTEGER MTOT, MERR	COMM	9
13	REAL MATTEN, MRCOE, MFREQ, QA, QR	COMM	10
14	CHARACTER * 3 MAT	COMM	11
15	CHARACTER * 70 MATDESC	COMM	12
16	*****	COMM	13
17	*****	COMM	14
18	INTEGER R,C	SMATDB	187
19	CHARACTER * 3 MATID	SMATDB	188
20	LOGICAL FOUND	SMATDB	189
21	OPEN (UNIT =6, FILE='MATTER', FORM='FORMATTED',	SMATDB	190
22	Z ACCESS='SEQUENTIAL', STATUS='NEW')	SMATDB	191
23	REWIND (6)	SMATDB	192
24	DO 10 R=1,MMAX	SMATDB	193
25	MATID = MAT(R)	SMATDB	194
26	IF (MATID(1:1) .EQ. 'M') THEN	SMATDB	195
27	WRITE (6,100) MAT(R)	SMATDB	196
28	WRITE (6,200) MATDESC (R)	SMATDB	197
29	WRITE (6,400) 1E+4, 1E+5 ,1E+6, 1E+7, 1E+8, 1E+9, 1E+10	SMATDB	198
30	WRITE (6,400) (MATTEN(R,C),C=1,7)	SMATDB	199
31	WRITE (6,400) QA(R)	SMATDB	200
32	WRITE (6,400) (MRCOE(R,C),C=1,7)	SMATDB	201
33	WRITE (6,400) QR(R)	SMATDB	202
34	ENDIF	SMATDB	203
35	10 CONTINUE	SMATDB	204
36	ENDFILE (6)	SMATDB	205
37	CALL PF ('REPLACE',0,'MATTER')	SMATDB	206
38	CLOSE (6, STATUS = 'DELETE')	SMATDB	207
39	100 FORMAT (A3)	SMATDB	208
40	200 FORMAT (A70)	SMATDB	209
41	400 FORMAT (7(1X, E9.3))	SMATDB	210
42	RETURN	SMATDB	211
43	END	SMATDB	212

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	270B		INTEGER	
FOUND	NONE	UNUSED/*S*	LOGICAL	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	271B		CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOE	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	267B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

MMAX INTEGER 100

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PF 3 SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	35
100	164B	FORMAT	39
200	166B	FORMAT	40
400	170B	FORMAT	41

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

QUIT 5B 0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	277B = 191
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.101 SECONDS

1		SMATDB	213
2		SMATDB	214
3		SMATDB	215
4		SMATDB	216
5	SUBROUTINE CHANGE	SMATDB	217
6	*****	COMM	1
7	*** COMMON FOR DATABASE OF MATERIAL PROPERTIES ***	COMM	2
8	*****	COMM	3
9	INTEGER MMAX	COMM	4
10	PARAMETER (MMAX=100)	COMM	5
11	COMMON /MATN/ MATTEN(MMAX,7), MRCOE(MMAX,7), QA(MMAX), QR(MMAX),	COMM	6
12	\$ MFREQ(MMAX,7), MERR, MTOT	COMM	7
13	COMMON /MATC/ MAT(MMAX), MATDESC(MMAX)	COMM	8
14	INTEGER MTOT, MERR	COMM	9
15	REAL MATTEN, MRCOE, MFREQ, QA, QR	COMM	10
16	CHARACTER * 3 MAT	COMM	11
17	CHARACTER * 70 MATDESC	COMM	12
18	*****	COMM	13
19	*****	COMM	14
20	INTEGER R,C,COMMAND, VAL, INDEX	SMATDB	219
21	CHARACTER * 3 MATID	SMATDB	220
22	LOGICAL FOUND	SMATDB	221
23	PRINT *, 'ENTER MATERIAL I.D. OF GROUP TO BE CHANGED'	SMATDB	222
24	READ *, MATID	SMATDB	223
25	FOUND = .FALSE.	SMATDB	224
26	DO 20 R = 1,MMAX	SMATDB	225
27	IF (MAT(R) .EQ. MATID) THEN	SMATDB	226
28	PRINT *	SMATDB	227
29	PRINT *, 'LINE 1: MATERIAL ID '	SMATDB	228
30	PRINT *, MAT (R)	SMATDB	229
31	PRINT *, 'LINE 2: DESCRIPTION'	SMATDB	230
32	PRINT *, MATDESC (R)	SMATDB	231
33	PRINT *, 'LINE 3: FREQUENCIES'	SMATDB	232
34	PRINT *, (MFREQ (R,C), C=1,7)	SMATDB	233
35	PRINT *, 'LINE 4: ATTENUATIONS'	SMATDB	234
36	PRINT *, (MATTEN (R,C), C=1,7)	SMATDB	235
37	PRINT *, 'LINE 5: ATTENUATION QUALITY PERCENT'	SMATDB	236
38	PRINT *, QA (R)	SMATDB	237
39	PRINT *, 'LINE 6: REFLECTION COEFFICIENTS'	SMATDB	238
40	PRINT *, (MRCOE (R,C), C=1,7)	SMATDB	239
41	PRINT *, 'LINE 7: REFLECTION QUALITY PERCENT'	SMATDB	240
42	PRINT *, QR (R)	SMATDB	241
43	FOUND = .TRUE.	SMATDB	242
44	INDEX = R	SMATDB	243
45	ENDIF	SMATDB	244
46	20 CONTINUE	SMATDB	245
47	IF (.NOT. FOUND) THEN	SMATDB	246
48	PRINT *, 'MATERIAL ', MATID, ' NOT FOUND'	SMATDB	247
49	RETURN	SMATDB	248
50	ENDIF	SMATDB	249
51	30 PRINT *	SMATDB	250
52	PRINT *, 'ENTER NUMBER OF LINE TO BE CHANGED',	SMATDB	251
53	Z ' (99 TO END CHANGES)'	SMATDB	252
54	READ *, COMMAND	SMATDB	253
55	IF (COMMAND .EQ. 1) THEN	SMATDB	254
56	PRINT *, 'ENTER NEW I.D.'	SMATDB	255
57	MAT (INDEX) = ' '	SMATDB	256
58	READ *, MATID	SMATDB	257
59	R = VAL (MATID(2:3))	SMATDB	258
60	MAT (R) = MATID	SMATDB	259
61	MATDESC (R) = MATDESC (INDEX)	SMATDB	260
62	MFREQ (R,1) = MFREQ (INDEX,1)	SMATDB	261
63	MFREQ (R,2) = MFREQ (INDEX,2)	SMATDB	262
64	MFREQ (R,3) = MFREQ (INDEX,3)	SMATDB	263

65	MFREQ (R,4) = MFREQ (INDEX,4)	SMATDB	264
66	MFREQ (R,5) = MFREQ (INDEX,5)	SMATDB	265
67	MFREQ (R,6) = MFREQ (INDEX,6)	SMATDB	266
68	MFREQ (R,7) = MFREQ (INDEX,7)	SMATDB	267
69	MATTEN (R,1) = MATTEN (INDEX,1)	SMATDB	268
70	MATTEN (R,2) = MATTEN (INDEX,2)	SMATDB	269
71	MATTEN (R,3) = MATTEN (INDEX,3)	SMATDB	270
72	MATTEN (R,4) = MATTEN (INDEX,4)	SMATDB	271
73	MATTEN (R,5) = MATTEN (INDEX,5)	SMATDB	272
74	MATTEN (R,6) = MATTEN (INDEX,6)	SMATDB	273
75	MATTEN (R,7) = MATTEN (INDEX,7)	SMATDB	274
76	QA (R) = QA (INDEX)	SMATDB	275
77	MRCOEF (R,1) = MRCOEF (INDEX,1)	SMATDB	276
78	MRCOEF (R,2) = MRCOEF (INDEX,2)	SMATDB	277
79	MRCOEF (R,3) = MRCOEF (INDEX,3)	SMATDB	278
80	MRCOEF (R,4) = MRCOEF (INDEX,4)	SMATDB	279
81	MRCOEF (R,5) = MRCOEF (INDEX,5)	SMATDB	280
82	MRCOEF (R,6) = MRCOEF (INDEX,6)	SMATDB	281
83	MRCOEF (R,7) = MRCOEF (INDEX,7)	SMATDB	282
84	QR (R) = QR (INDEX)	SMATDB	283
85	INDEX = R	SMATDB	284
86	ELSE IF (COMMAND .EQ. 2) THEN	SMATDB	285
87	PRINT *, 'ENTER NEW ONE LINE DESCRIPTION OF MATERIAL'	SMATDB	286
88	READ *, MATDESC (INDEX)	SMATDB	287
89	ELSE IF (COMMAND .EQ. 3) THEN	SMATDB	288
90	PRINT *, 'ENTER NEW SET OF 7 FREQUENCIES'	SMATDB	289
91	READ *, (MFREQ (INDEX,C), C=1,7)	SMATDB	290
92	ELSE IF (COMMAND .EQ. 4) THEN	SMATDB	291
93	PRINT *, 'ENTER NEW SET OF 7 ATTENUATIONS'	SMATDB	292
94	READ *, (MATTEN (INDEX,C), C=1,7)	SMATDB	293
95	ELSE IF (COMMAND .EQ. 5) THEN	SMATDB	294
96	PRINT *, 'ENTER NEW ATTENUATION QUALITY PERCENT'	SMATDB	295
97	READ *, QA (INDEX)	SMATDB	296
98	ELSE IF (COMMAND .EQ. 6) THEN	SMATDB	297
99	PRINT *, 'ENTER NEW SET OF 7 REFLECTION COEFFICIENTS'	SMATDB	298
100	READ *, (MRCOEF (INDEX,C), C=1,7)	SMATDB	299
101	ELSE IF (COMMAND .EQ. 7) THEN	SMATDB	300
102	PRINT *, 'ENTER NEW REFLECTION QUALITY PERCENT'	SMATDB	301
103	READ *, QR (INDEX)	SMATDB	302
104	ELSE IF (COMMAND .EQ. 99) THEN	SMATDB	303
105	GOTO 40	SMATDB	304
106	ENDIF	SMATDB	305
107	GOTO 30	SMATDB	306
108 C		SMATDB	307
109 40	PRINT *	SMATDB	308
110	PRINT *, MAT (INDEX)	SMATDB	309
111	PRINT *, MATDESC (INDEX)	SMATDB	310
112	PRINT *, (MFREQ (INDEX,C), C=1,7)	SMATDB	311
113	PRINT *, (MATTEN (INDEX,C), C=1,7)	SMATDB	312
114	PRINT *, QA (INDEX)	SMATDB	313
115	PRINT *, (MRCOEF (INDEX,C), C=1,7)	SMATDB	314
116	PRINT *, QR (INDEX)	SMATDB	315
117	PRINT *	SMATDB	316
118	RETURN	SMATDB	317
119	END	SMATDB	318

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	1137B		INTEGER	
COMMAND	1140B		INTEGER	
FOUND	1143B		LOGICAL	
INDEX	1141B		INTEGER	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	1142B		CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOEFF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	1136B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

MMAX	INTEGER	100
------	---------	-----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

VAL	INTEGER	1	FUNCTION
-----	---------	---	----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

20	INACTIVE	DO-TERM	46
30	204B		51
40	477B		109

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

CHANGE	5B	0
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	1160B = 624
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.318 SECONDS

```

1 SUBROUTINE LMATTER                                LMATTER
2 *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LMATTER
3 *!!!                                              !!!LMATTER
4 *!!!   THIS SUBROUTINE LOADS THE MATERIAL DATABASE INTO ARRAYS FOR !!!LMATTER
5 *!!!   FURTHER PROGRAM USE.                      !!!LMATTER
6 *!!!                                              !!!LMATTER
7 *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LMATTER
8 *****LMATTER
9 *****COMM
10 *** COMMON FOR DATABASE OF MATERIAL PROPERTIES ***COMM
11 *****COMM
12 INTEGER MMAX                                     COMM
13 PARAMETER (MMAX=100)                             COMM
14 COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM
15 $ MFREQ(MMAX,7), MERR, MTOT                       COMM
16 COMMON /MATC/MAT(MMAX),MATDESC(MMAX)             COMM
17 INTEGER MTOT, MERR                               COMM
18 REAL MATTEN, MRCOEF, MFREQ, QA, QR               COMM
19 CHARACTER * 3 MAT                                COMM
20 CHARACTER * 70 MATDESC                           COMM
21 *****COMM
22 *****COMM
23 *****
24 * DECLARATION OF VARIABLES                        LMATTER
25 *****LMATTER
26 INTEGER R, C, VAL                                LMATTER
27 CHARACTER * 3 MATID                              LMATTER
28 *****LMATTER
29 * GET FILE                                        LMATTER
30 *****LMATTER
31 MERR = 0                                          LMATTER
32 CALL PF ('GET',0,'MATTER','RC',MERR)            LMATTER
33 *****LMATTER
34 * FILE ERROR CHECK                               LMATTER
35 *****LMATTER
36 IF ( MERR.EQ. 0 ) THEN                          LMATTER
37 999 CONTINUE                                    LMATTER
38 ELSE IF ( MERR.EQ. 2 ) THEN                     LMATTER
39 CALL WARNING (3)                                LMATTER
40 RETURN                                           LMATTER
41 ELSE                                             LMATTER
42 CALL WARNING (4)                                LMATTER
43 RETURN                                           LMATTER
44 END IF                                           LMATTER
45 *****LMATTER
46 * OPEN FILE                                       LMATTER
47 *****LMATTER
48 OPEN (UNIT = 3, FILE='MATTER',FORM='FORMATTED', LMATTER
49 $ STATUS = 'OLD', ACCESS = 'SEQUENTIAL')        LMATTER
50 REWIND (3)                                       LMATTER
51 *****LMATTER
52 * INITIALIZE ARRAYS                             LMATTER
53 *****LMATTER
54 DATA MAT / 100 * ' ' /                         LMATTER
55 DATA MATDESC / 100 * ' ' /                    LMATTER
56 DATA MFREQ / 700 * 0.0 /                       LMATTER
57 DATA MATTEN / 700 * 0.0 /                      LMATTER
58 DATA QA / 100 * 0.0 /                          LMATTER
59 DATA MRCOEF / 700 * 0.0 /                      LMATTER
60 DATA QR / 100 * 0.0 /                          LMATTER
61 *****LMATTER
62 * READ IN THE MATERIAL FILE                     LMATTER
63 *****LMATTER
64 10 READ(3,1000,END=20) MATID                   LMATTER

```

65	R = VAL(MATID(2:3))	LMATTER	52
66	MAT (R) = MATID	LMATTER	53
67	READ (3,2000,END=20) MATDESC (R)	LMATTER	54
68	READ (3,4000,END=20) (MFREQ(R,C),C=1,7)	LMATTER	55
69	READ (3,4000,END=20) (MATTEN(R,C),C=1,7)	LMATTER	56
70	READ (3,4000,END=20) QA (R)	LMATTER	57
71	READ (3,4000,END=20) (MRCOEF(R,C),C=1,7)	LMATTER	58
72	READ (3,4000,END=20) QR (R)	LMATTER	59
73	GOTO 10	LMATTER	60
74	20 CONTINUE	LMATTER	61
75	1000 FORMAT (A3)	LMATTER	62
76	2000 FORMAT (A70)	LMATTER	63
77	4000 FORMAT (7(1X,E9.3))	LMATTER	64
78	*****	LMATTER	65
79	* CLOSE FILE	LMATTER	66
80	*****	LMATTER	67
81	CLOSE (3,STATUS = 'DELETE')	LMATTER	68
82	RETURN	LMATTER	69
83	END	LMATTER	70

--VARIABLE MAP--(LO=A)
--NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	327B		INTEGER	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	330B		CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOEF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	326B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)
--NAME---TYPE-----VALUE

MMAI	INTEGER	100
------	---------	-----

--PROCEDURES--(LO=A)
--NAME-----TYPE-----ARGS-----CLASS-----

PF		5	SUBROUTINE
VAL	INTEGER	1	FUNCTION
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)
--LABEL-ADDRESS-----PROPERTIES-----DEF --LABEL-ADDRESS-----PROPERTIES-----DEF

10	36B	64	1000	202B	FORMAT	75
20	160B	74	2000	204B	FORMAT	76
999	*NO REFS*	37	4000	206B	FORMAT	77

FTN 5.1+552 83/12/24. 09.48.06 PAGE 17
SUBROUTINE LMATTER 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

LMATTER 5B 0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE3 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	336B = 222
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.127 SECONDS

1	SUBROUTINE WARNING(ERR)	WARNING	1
2	INTEGER ERR, ERRM	WARNING	2
3	CHARACTER*45 MESSAGE(20)	WARNING	3
4	DATA MESSAGE(1)/'HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	4
5	DATA MESSAGE(2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE' /	WARNING	5
6	DATA MESSAGE(3)/'MATTER" FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	6
7	DATA MESSAGE(4)/'FILE HANDLING PROBLEM ON "MATTER" FILE' /	WARNING	7
8	DATA MESSAGE(5)/'TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	8
9	DATA MESSAGE(6)/'FILE HANDLING PROBLEM ON "TYPE" FILE' /	WARNING	9
10	DATA MESSAGE(7)/'WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	10
11	DATA MESSAGE(8)/'FILE HANDLING PROBLEM ON "WALL" FILE' /	WARNING	11
12	DATA MESSAGE(9)/'HEIGHT AND WIDTH OF ROOM MISSING' /	WARNING	12
13	DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING' /	WARNING	13
14	DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	14
15	DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE' /	WARNING	15
16	DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE' /	WARNING	16
17	DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE' /	WARNING	17
18	DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE' /	WARNING	18
19	DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE' /	WARNING	19
20	DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE' /	WARNING	20
21	DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE' /	WARNING	21
22	DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE' /	WARNING	22
23	DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE' /	WARNING	23
24	ERRM=12	WARNING	24
25	IERR = ERR	WARNING	25
26	IF(ERR.GT.ERRM) IERR=20	WARNING	26
27	WRITE(6,20)	WARNING	27
28	WRITE(6,10) ERR,MESSAGE(IERR)	WARNING	28
29	WRITE(6,20)	WARNING	29
30 10	FORMAT(' ***WARNING NUMBER = ',15,' *** ',A45)	WARNING	30
31 20	FORMAT(' ')	WARNING	31
32	RETURN	WARNING	32
33	END	WARNING	33

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ERR	1	DUMMY-ARG	INTEGER	
ERRM	60B		INTEGER	
IERR	213B		INTEGER	
MESSAGE	61B		CHAR*45	20

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS---PROPERTIES---DEF

10	34B	FORMAT	30
20	42B	FORMAT	31

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

WARNING	5B	1
---------	----	---

FTN 5.1+552 83/12/24. 09.48.06 PAGE 19
 SUBROUTINE WARNING 74/175 OPT=0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	216B = 142
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.063 SECONDS

1	SUBROUTINE ERROR(IERR)	ERROR	1
2	CHARACTER*45 MESSAGE(20)	ERROR	2
3	DATA MESSAGE(1)/'MATERIALS DATA BASE IS EMPTY	// ERROR	3
4	DATA MESSAGE(2)/'FREQUENCY IS OUT OF RANGE	// ERROR	4
5	DATA MESSAGE(3)/'THIS MATERIAL IS NOT IN DATA BASE	// ERROR	5
6	DATA MESSAGE(4)/'DENOMINATOR IS ZERO	// ERROR	6
7	DATA MESSAGE(5)/'FILE HANDLING ERROR	// ERROR	7
8	DATA MESSAGE(6)/'ERROR CODE IS OUT OF RANGE	// ERROR	8
9	DATA MESSAGE(7)/'ERROR CODE IS OUT OF RANGE	// ERROR	9
10	DATA MESSAGE(8)/'ERROR CODE IS OUT OF RANGE	// ERROR	10
11	DATA MESSAGE(9)/'ERROR CODE IS OUT OF RANGE	// ERROR	11
12	DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE	// ERROR	12
13	DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE	// ERROR	13
14	DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE	// ERROR	14
15	DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE	// ERROR	15
16	DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE	// ERROR	16
17	DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE	// ERROR	17
18	DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE	// ERROR	18
19	DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE	// ERROR	19
20	DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE	// ERROR	20
21	DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE	// ERROR	21
22	DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE	// ERROR	22
23	IERRM=5	ERROR	23
24	IF(IERR.GT.IERRM) IERR=20	ERROR	24
25	WRITE(6,10) IERR,MESSAGE(IERR)	ERROR	25
26	10 FORMAT(' ***ERROR NUMBER = ',15,' *** ',A45)	ERROR	26
27	CALL PMDSTOP	ERROR	27
28	STOP 'ERROR'	ERROR	28
29	END	ERROR	29

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

IERR	1	DUMMY-ARG	INTEGER	
IERRM	210B		INTEGER	
MESSAGE	56B		CHAR*45	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PMDSTOP	0	SUBROUTINE
---------	---	------------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS---PROPERTIES---DEF

10	36B	FORMAT	26
----	-----	--------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

ERROR	5B	1
-------	----	---

FTN 5.1+552 83/12/24. 09.48.06 PAGE 21
SUBROUTINE ERROR 74/175 OPT=0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE 6	FMT/SEQ
--------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	213B = 139
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.056 SECONDS

Appendix 9.3 Listing of Computer Program SHOLES

```

1      PROGRAM SHOLES (INPUT,TAPE1=INPUT)                                SHOLES 1
2      *                                                                    SHOLES 2
3      *THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH HOLE    SHOLES 3
4      *IN THE BUILDING AND STORES IT. THE FILE NAME IS CREATED BY        SHOLES 4
5      *ATTACHING "B" TO THE FRONT OF AND "H" TO THE BACK OF THE BUILDING  SHOLES 5
6      *IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE        SHOLES 6
7      *THAN 5 ALPHANUMERIC CHARACTERS.                                  SHOLES 7
8                                                                    SHOLES 8
9      *****COMF 1
10     *** COMMON FOR INITIAL PARAMETERS                                ***COMF 2
11     *****COMF 3
12     INTEGER FMAX                                                    COMF 4
13     PARAMETER (FMAX = 50)                                           COMF 5
14     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF 6
15     $ FTOT                                                         COMF 7
16     COMMON /INITILC/ BLDG                                           COMF 8
17     CHARACTER * 5 BLDG                                             COMF 9
18     REAL FREQ, AFLAG, RFLAG, FREQA                                COMF 10
19     INTEGER QUALITY, FERR, FTOT                                    COMF 11
20     *****COMF 12
21     *****COMF 13
22     *****COMR 1
23     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS        ***COMR 2
24     *****COMR 3
25     INTEGER RMAX                                                    COMR 4
26     PARAMETER (RMAX = 20)                                           COMR 5
27     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)    COMR 6
28     INTEGER NROOMS                                                 COMR 7
29     REAL ROOM                                                       COMR 8
30     *****COMR 9
31     *****COMR 10
32     *****COMH 1
33     *** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS        ***COMH 2
34     *****COMH 3
35     INTEGER HMAX                                                    COMH 4
36     PARAMETER (HMAX = 35)                                           COMH 5
37     COMMON /HOLEN/ HTOT, HERR                                       COMH 6
38     COMMON /HOLEC/ HOLE(HMAX,4)                                     COMH 7
39     INTEGER HTOT, HERR                                              COMH 8
40     CHARACTER * 3 HOLE                                             COMH 9
41     * ===== COMH 10
42     * DESCRIPTION OF ARRAYS                                           COMH 11
43     * ===== COMH 12
44     * ROOM IDENTIFICATION APERTURE ID COMH 13
45     * ----- COMH 14
46     * DIRECTION FROM ROOM TO ROOM COMH 15
47     * ----- COMH 16
48     * HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4) COMH 17
49     * A3 A3 A3 A3 COMH 18
50     *****COMH 19
51     *****COMH 20
52     INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE          SHOLES 12
53     INTEGER IERR                                                    SHOLES 13
54     CHARACTER * 7 PFN                                              SHOLES 14
55     *                                                                SHOLES 15
56     * INITIALIZATION SHOLES 16
57     QUIT = 0 SHOLES 17
58     HTOT = 0 SHOLES 18
59     ABORT = 0 SHOLES 19
60 100 PRINT* SHOLES 20
61     PRINT *, 'ENTER BUILDING IDENTIFICATION (E.G. '101')' SHOLES 21
62     PRINT *, ' (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)' SHOLES 22
63     REWIND 1 SHOLES 23
64     READ(1,*,END=100) BLDG SHOLES 24

```

65		SHOLES	25
66	IF (GETLEN(BLDG) .GT. 5) THEN	SHOLES	26
67	GO TO 100	SHOLES	27
68	END IF	SHOLES	28
69	PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'H'	SHOLES	29
70	*	SHOLES	30
71	*** LOAD DATA FROM EXISTING FILE IF NECESSARY	SHOLES	31
72	200 PRINT*	SHOLES	32
73	PRINT*, 'WILL THIS BE'	SHOLES	33
74	PRINT*, ' (1) A MODIFICATION OF AN EXISTING FILE?'	SHOLES	34
75	PRINT*, ' (2) A NEW FILE?'	SHOLES	35
76	PRINT*, 'ENTER A NUMBER !!!'	SHOLES	36
77	REWIND 1	SHOLES	37
78	READ(1,*,END=200) OLDFILE	SHOLES	38
79	IF ((OLDFILE .NE. 1) .AND. (OLDFILE .NE. 2)) THEN	SHOLES	39
80	GOTO 200	SHOLES	40
81	ELSE IF (OLDFILE .EQ. 1) THEN	SHOLES	41
82	*	SHOLES	42
83	*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	SHOLES	43
84	IERR = 0	SHOLES	44
85	CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)	SHOLES	45
86	IF (IERR .EQ. 2) THEN	SHOLES	46
87	PRINT*	SHOLES	47
88	PRINT *, 'FILE ',PFN, ' NOT FOUND'	SHOLES	48
89	PRINT*, 'PROGRAM ABORTED!!!'	SHOLES	49
90	PRINT*	SHOLES	50
91	PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART '	SHOLES	51
92	+ 'PROGRAM'	SHOLES	52
93	PRINT*	SHOLES	53
94	STOP	SHOLES	54
95	*	SHOLES	55
96	ELSE	SHOLES	56
97	CALL LHOLE	SHOLES	57
98	IF (HERR .NE. 0) CALL ERROR(5)	SHOLES	58
99	END IF	SHOLES	59
100	ELSE IF (OLDFILE .EQ. 2) THEN	SHOLES	60
101	*	SHOLES	61
102	*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	SHOLES	62
103	IERR = 0	SHOLES	63
104	CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)	SHOLES	64
105	IF (IERR .EQ. 0) THEN	SHOLES	65
106	PRINT*	SHOLES	66
107	PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ',BLDG	SHOLES	67
108	PRINT*	SHOLES	68
109	PRINT*, 'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE '	SHOLES	69
110	+ 'OVER THE OLD FILE.'	SHOLES	70
111	250 PRINT*	SHOLES	71
112	PRINT*, 'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'	SHOLES	72
113	PRINT*, 'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'	SHOLES	73
114	REWIND 1	SHOLES	74
115	READ(1,*,END=250) ANSWER	SHOLES	75
116	IF (ANSWER .EQ. 1) THEN	SHOLES	76
117	PRINT*	SHOLES	77
118	PRINT*, 'PROGRAM HAS BEEN ABORTED, AT YOUR REQUEST'	SHOLES	78
119	PRINT*	SHOLES	79
120	STOP	SHOLES	80
121	ELSE IF (ANSWER .EQ. 2) THEN	SHOLES	81
122	CONTINUE	SHOLES	82
TRIVIAL*	CONTINUE WITH NO STATEMENT LABEL -- IGNORED		
123	ELSE	SHOLES	83
124	GOTO 250	SHOLES	84
125	END IF	SHOLES	85
126	ELSE IF (IERR .EQ. 2) THEN	SHOLES	86
127	*	SHOLES	87

128 *** NO DATA FILE ALREADY EXISTS FOR THIS BUILDING AND DATA ENTRY	SHOLES	88
129 *** CAN CONTINUE	SHOLES	89
130 CONTINUE	SHOLES	90
TRIVIAL* CONTINUE WITH NO STATEMENT LABEL -- IGNORED		
131 ELSE	SHOLES	91
132 *	SHOLES	92
133 *** PERMANENT FILE ERROR	SHOLES	93
134 PRINT*	SHOLES	94
135 PRINT*, 'PROGRAM ABORTED !!!'	SHOLES	95
136 PRINT*, ' SOME PERMANENT FILE ERROR HAS OCCURRED.'	SHOLES	96
137 PRINT*, ' DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',	SHOLES	97
138 + 'AND TRY AGAIN'	SHOLES	98
139 STOP	SHOLES	99
140 END IF	SHOLES	100
141 *	SHOLES	101
142 PRINT*	SHOLES	102
143 PRINT*, ' BEGIN ENTERING DATA'	SHOLES	103
144 300 HTOT = HTOT + 1	SHOLES	104
145 IF (HTOT .EQ. 1) THEN	SHOLES	105
146 CALL DATAIN(1,HTOT)	SHOLES	106
147 ELSE	SHOLES	107
148 CALL DATAIN (0,HTOT)	SHOLES	108
149 END IF	SHOLES	109
150 400 PRINT*	SHOLES	110
151 PRINT*, 'DO YOU WANT TO ENTER MORE DATA?'	SHOLES	111
152 + '(1) YES (2) NO'	SHOLES	112
153 PRINT*, ' ENTER A NUMBER !!!'	SHOLES	113
154 REWIND 1	SHOLES	114
155 READ(1,*,END=400) ANSWER	SHOLES	115
156 IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SHOLES	116
157 GOTO 400	SHOLES	117
158 ELSE IF (ANSWER .EQ. 1) THEN	SHOLES	118
159 GOTO 300	SHOLES	119
160 ELSE IF (ANSWER .EQ. 2) THEN	SHOLES	120
161 PRINT*	SHOLES	121
162 PRINT*, 'DATA ENTRY DISCONTINUED'	SHOLES	122
163 END IF	SHOLES	123
164 END IF	SHOLES	124
165 *	SHOLES	125
166 *** MANIPULATE DATA	SHOLES	126
167 CALL MANIP (QUIT,ABORT)	SHOLES	127
168 *	SHOLES	128
169 *** TERMINATE PROGRAM, STORING DATA IF NECESSARY	SHOLES	129
170 IF (QUIT .EQ. 1) THEN	SHOLES	130
171 OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',	SHOLES	131
172 + ACCESS='SEQUENTIAL',STATUS='NEW')	SHOLES	132
173 500 FORMAT (1X,4(1X,A3))	SHOLES	133
174 DO 600 N = 1,HTOT	SHOLES	134
175 WRITE (6,500)(HOLE(N,Y1), Y1=1,4)	SHOLES	135
176 600 CONTINUE	SHOLES	136
177 ENDFILE(6)	SHOLES	137
178 CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))	SHOLES	138
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT		
179 CLOSE(6,STATUS='DELETE')	SHOLES	139
180 PRINT*	SHOLES	140
181 PRINT*, 'DATA HAS BEEN STORED AND PROGRAM TERMINATED'	SHOLES	141
182 END IF	SHOLES	142
183 IF(ABORT .EQ. 1) THEN	SHOLES	143
184 PRINT*	SHOLES	144
185 PRINT*, 'PROGRAM HAS BEEN ABORTED'	SHOLES	145
186 PRINT*, ' NO DATA HAS BEEN STORED !!!'	SHOLES	146
187 END IF	SHOLES	147
188 STOP	SHOLES	148
189 END	SHOLES	149

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ABORT	1054B			INTEGER	
AFLAG	2B	/INITILN/		REAL	
ANSWER	1055B			INTEGER	
BLDG	0B	/INITILC/		CHAR*5	
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREGA	4B	/INITILN/		REAL	50
FTOT	67B	/INITILN/		INTEGER	
HERR	1B	/HOLEN/		INTEGER	
HOLE	0B	/HOLEC/		CHAR*3	140
HTOT	0B	/HOLEN/		INTEGER	
IERR	1061B			INTEGER	
LINE	NONE		UNUSED/*S*	INTEGER	
N	1057B			INTEGER	
NROOMS	1244B	/ROOMN/		INTEGER	
OLDFILE	1056B			INTEGER	
PFN	1062B			CHAR*7	
QUALITY	1B	/INITILN/		INTEGER	
QUIT	1053B			INTEGER	
RAREA	1245B	/ROOMN/		REAL	20
RFLAG	3B	/INITILN/		REAL	
ROOM	0B	/ROOMN/		REAL	676
Y1	1060B			INTEGER	
Y2	NONE		UNUSED/*S*	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

NAME	TYPE	VALUE
FMAX	INTEGER	50
HMAX	INTEGER	35
RMAX	INTEGER	20

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS	NAME	TYPE	ARGS	CLASS
DATAIN		2	SUBROUTINE	LHOLE		0	SUBROUTINE
ERROR		1	SUBROUTINE	MANIP		2	SUBROUTINE
GETLEN	INTEGER	1	FUNCTION	PF		5	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF	LABEL	ADDRESS	PROPERTIES	DEF
100	21B		60	400	256B		150
200	47B		72	500	606B	FORMAT	173
250	166B		111	600	INACTIVE	DO-TERM	176
300	244B		144				

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
SHOLES	14B	0

FTN 5.1+552 83/12/24. 09.11.45 PAGE 5
PROGRAM SHOLES 74/175 OPT=0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE1 FMT/SEQ
TAPE6 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	1065B = 565
CM LABELLED COMMON LENGTH	1436B = 798
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.270 SECONDS

2 TRIVIAL ERRORS IN SHOLES

1 WARNING ERROR IN SHOLES

1	SUBROUTINE DATAIN (INSERT,LINE)	SHOLES	150
2	*****	COMR	1
3	*** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS	***COMR	2
4	*****	COMR	3
5	INTEGER RMAX	COMR	4
6	PARAMETER (RMAX = 20)	COMR	5
7	COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)	COMR	6
8	INTEGER NROOMS	COMR	7
9	REAL ROOM	COMR	8
10	*****	COMR	9
11	*****	COMR	10
12	*****	COMH	1
13	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS	***COMH	2
14	*****	COMH	3
15	INTEGER HMAX	COMH	4
16	PARAMETER (HMAX = 35)	COMH	5
17	COMMON /HOLEN/ HTOT, HERR	COMH	6
18	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
19	INTEGER HTOT, HERR	COMH	8
20	CHARACTER * 3 HOLE	COMH	9
21	* =====	COMH	10
22	* DESCRIPTION OF ARRAYS	COMH	11
23	* =====	COMH	12
24	* ROOM IDENTIFICATION APERTURE ID	COMH	13
25	* -----	COMH	14
26	* DIRECTION FROM ROOM TO ROOM	COMH	15
27	* -----	COMH	16
28	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17
29	* A3 A3 A3 A3	COMH	18
30	*****	COMH	19
31	*****	COMH	20
32	INTEGER ANSWER,LOK,DOK,NOK,GETLEN,VAL, INSERT,LINE,V	SHOLES	153
33	CHARACTER *3 DIR,FROM,TO,ID	SHOLES	154
34	200 PRINT*	SHOLES	155
35	PRINT*, 'ENTER DIRECTION (E. G. 'LR')'	SHOLES	156
36	REWIND 1	SHOLES	157
37	READ(1,*,END=200) DIR	SHOLES	158
38	IF ((DIR .NE. 'LR')	SHOLES	159
39	+ .AND. (DIR .NE. 'FB')	SHOLES	160
40	+ .AND. (DIR .NE. 'UD')) THEN	SHOLES	161
41	PRINT*, 'DIRECTION MUST BE 'LR' OR 'FB' OR 'UD''	SHOLES	162
42	PRINT*, 'TRY AGAIN!!!'	SHOLES	163
43	GOTO 200	SHOLES	164
44	END IF	SHOLES	165
45	HOLE(LINE,1) = DIR	SHOLES	166
46	*	SHOLES	167
47	300 PRINT*	SHOLES	168
48	PRINT*, 'ENTER "FROM" (E.G. '02' OR 'D1')'	SHOLES	169
49	REWIND 1	SHOLES	170
50	READ(1,*,END=300) FROM	SHOLES	171
51	LOK = 0	SHOLES	172
52	DOK = 0	SHOLES	173
53	NOK = 0	SHOLES	174
54	IF (GETLEN(FROM) .EQ. 2) THEN	SHOLES	175
55	LOK = 1	SHOLES	176
56	END IF	SHOLES	177
57	IF (FROM(1:1) .EQ. 'D') THEN	SHOLES	178
58	V = VAL(FROM(2:2))	SHOLES	179
59	IF ((V .GE. 1) .AND. (V .LE. 6)) THEN	SHOLES	180
60	DOK = 1	SHOLES	181
61	END IF	SHOLES	182
62	END IF	SHOLES	183
63	IF ((ICHR(FROM(1:1)) .GE. 16)	SHOLES	184
64	+ .AND. (ICHR(FROM(1:1)) .LE. 25)	SHOLES	185

65	+	.AND. (ICAR(FROM(2:2)) .GE. 16)	SHOLES	186
66	+	.AND. (ICAR(FROM(2:2)) .LE. 25)	SHOLES	187
67	+	.AND. (GETLEN(FROM) .EQ. 2)) THEN	SHOLES	188
68		V = VAL(FROM)	SHOLES	189
69		IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN	SHOLES	190
70		NOK = 1	SHOLES	191
71		END IF	SHOLES	192
72		END IF	SHOLES	193
73		IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1) .OR. (NOK .EQ. 1))) THEN	SHOLES	194
74		HOLE(LINE,2) = FROM	SHOLES	195
75		ELSE	SHOLES	196
76		PRINT*	SHOLES	197
77		PRINT*, 'INCORRECT ENTRY. TRY AGAIN!!'	SHOLES	198
78		GOTO 300	SHOLES	199
79		END IF	SHOLES	200
80	*	SHOLES	201
81	400	PRINT*	SHOLES	202
82		PRINT*, 'ENTER "TO" (E.G. '02' OR 'D1')'	SHOLES	203
83		REWIND 1	SHOLES	204
84		READ(1,*,END=400) TO	SHOLES	205
85		LOK = 0	SHOLES	206
86		DOK = 0	SHOLES	207
87		NOK = 0	SHOLES	208
88		IF (GETLEN(TO) .EQ. 2) THEN	SHOLES	209
89		LOK = 1	SHOLES	210
90		END IF	SHOLES	211
91		IF (TO(1:1) .EQ. 'D') THEN	SHOLES	212
92		V = VAL(TO(2:2))	SHOLES	213
93		IF ((V .GE. 1) .AND. (V .LE. 6)) THEN	SHOLES	214
94		DOK = 1	SHOLES	215
95		END IF	SHOLES	216
96		END IF	SHOLES	217
97		IF ((ICAR(TO(1:1)) .GE. 16)	SHOLES	218
98	+	.AND. (ICAR(TO(1:1)) .LE. 25)	SHOLES	219
99	+	.AND. (ICAR(TO(2:2)) .GE. 16)	SHOLES	220
100	+	.AND. (ICAR(TO(2:2)) .LE. 25)	SHOLES	221
101	+	.AND. (GETLEN(TO) .EQ. 2)) THEN	SHOLES	222
102		V = VAL(TO)	SHOLES	223
103		IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN	SHOLES	224
104		NOK = 1	SHOLES	225
105		END IF	SHOLES	226
106		END IF	SHOLES	227
107		IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1) .OR. (NOK .EQ. 1))) THEN	SHOLES	228
108		HOLE(LINE,3) = TO	SHOLES	229
109		ELSE	SHOLES	230
110		PRINT*	SHOLES	231
111		PRINT*, 'INCORRECT ENTRY. TRY AGAIN!!'	SHOLES	232
112		GOTO 400	SHOLES	233
113		END IF	SHOLES	234
114		IF(FROM .EQ. TO) THEN	SHOLES	235
115		PRINT*	SHOLES	236
116		PRINT*, 'INCORRECT ENTRY!!'	SHOLES	237
117		PRINT*, '"FROM" CANNOT EQUAL "TO"'	SHOLES	238
118		PRINT*, 'CHECK YOUR DATA AND REENTER "FROM" AND "TO"'	SHOLES	239
119		PRINT*	SHOLES	240
120		GOTO 300	SHOLES	241
121		END IF	SHOLES	242
122		IF ((FROM(1:1) .EQ. 'D') .AND. (TO(1:1) .EQ. 'D')) THEN	SHOLES	243
123		PRINT*	SHOLES	244
124		PRINT*, 'INCORRECT ENTRY!!'	SHOLES	245
125		PRINT*, '"FROM" AND "TO" CANNOT BOTH CONTAIN "D"'	SHOLES	246
126		PRINT*, 'CHECK YOUR DATA AND REENTER "FROM" AND "TO"'	SHOLES	247
127		PRINT*	SHOLES	248
128		GOTO 300	SHOLES	249

129	END IF	SHOLES	250
130 *	SHOLES	251
131 500	PRINT*	SHOLES	252
132	PRINT*, 'ENTER HOLE 'ID' (E.G. 'WA' OR 'DA')'	SHOLES	253
133	REWIND 1	SHOLES	254
134	READ (1,*,END=500) ID	SHOLES	255
135	IF (((ID(1:1) .EQ. 'D') .OR. (ID(1:1) .EQ. 'W'))	SHOLES	256
136	+ .AND. (ICHAR(ID(2:2)) .LE. 58)	SHOLES	257
137	+ .AND. (ICHAR(ID(2:2)) .GE. 33)) THEN	SHOLES	258
138	HOLE(LINE,4) = ID	SHOLES	259
139	ELSE	SHOLES	260
140	GOTO 500	SHOLES	261
141	END IF	SHOLES	262
142 *	SHOLES	263
143	RETURN	SHOLES	264
144	END	SHOLES	265

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ANSWER	NONE	UNUSED/*S*	INTEGER	
DIR	727B		CHAR*3	
DOK	724B		INTEGER	
FROM	730B		CHAR*3	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
ID	732B		CHAR*3	
INSERT	1	DUMMY-ARG UNUSED	INTEGER	
LINE	2	DUMMY-ARG	INTEGER	
LOK	723B		INTEGER	
NOK	725B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
ROOM	0B	/ROOMN/	REAL	676
TO	731B		CHAR*3	
V	726B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

HMAX	INTEGER	35
RMAX	INTEGER	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC
VAL	INTEGER	1	FUNCTION

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

200	7B	34
300	46B	47
400	203B	81
500	402B	131

FTN 5.1+552 83/12/24. 09.11.45 PAGE 9
SUBROUTINE DATAIN 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS---ARGS---

DATAIN 5B 2

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE1 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	735B = 477
CM LABELLED COMMON LENGTH	1345B = 741
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.257 SECONDS

1	SUBROUTINE MANIP (QUIT,ABORT)	SHOLES	266
2	*****	COMH	1
3	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS	***COMH	2
4	*****	COMH	3
5	INTEGER HMAX	COMH	4
6	PARAMETER (HMAX = 35)	COMH	5
7	COMMON /HOLEN/ HTOT, HERR	COMH	6
8	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
9	INTEGER HTOT, HERR	COMH	8
10	CHARACTER * 3 HOLE	COMH	9
11	* =====	COMH	10
12	* DESCRIPTION OF ARRAYS	COMH	11
13	* =====	COMH	12
14	* ROOM IDENTIFICATION APERTURE ID	COMH	13
15	* -----	COMH	14
16	* DIRECTION FROM ROOM TO ROOM	COMH	15
17	* -----	COMH	16
18	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17
19	* A3 A3 A3 A3	COMH	18
20	*****	COMH	19
21	*****	COMH	20
22	INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT	SHOLES	268
23	INTEGER TEMP,V,X,Y,COMMAND	SHOLES	269
24	CHARACTER * 3 DIR, FROM, TO	SHOLES	270
25	*	SHOLES	271
26	10 FLAG1 = 0	SHOLES	272
27	PRINT*	SHOLES	273
28	PRINT*, 'CHOOSE'	SHOLES	274
29	PRINT*, ' (1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES'	SHOLES	275
30	PRINT*, ' (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'	SHOLES	276
31	PRINT*, ' (3) DELETE LINE (6) STORE DATA AND EXIT ',	SHOLES	277
32	+ 'PROGRAM'	SHOLES	278
33	PRINT*, ' (7) EXIT PROGRAM WITHOUT ',	SHOLES	279
34	+ 'STORING DATA'	SHOLES	280
35	PRINT*, 'ENTER A NUMBER !!!'	SHOLES	281
36	PRINT*	SHOLES	282
37	REWIND 1	SHOLES	283
38	READ(1,*,END=10) COMMAND	SHOLES	284
39	*	SHOLES	285
40	*-----	SHOLES	286
41	*** DISPLAY LINE ***	SHOLES	287
42	*-----	SHOLES	288
43	IF (COMMAND .EQ. 1) THEN	SHOLES	289
44	*	SHOLES	290
45	*** INDICATE EMPTY DATA FILE	SHOLES	291
46	IF (HTOT .EQ. 0) THEN	SHOLES	292
47	PRINT*	SHOLES	293
48	PRINT*, 'DATA FILE IS EMPTY !!!'	SHOLES	294
49	*	SHOLES	295
50	*** ENTER NUMBER OF LINE TO BE DISPLAYED	SHOLES	296
51	ELSE	SHOLES	297
52	100 PRINT*	SHOLES	298
53	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'	SHOLES	299
54	PRINT*, ' (ENTER "0" TO ESCAPE DISPLAY MODE) '	SHOLES	300
55	REWIND 1	SHOLES	301
56	READ(1,*,END=100) N	SHOLES	302
57	*	SHOLES	303
58	*** CHECK VALIDITY OF LINE NUMBER	SHOLES	304
59	IF ((N .GT. HTOT) .OR. (N .LT. 0)) THEN	SHOLES	305
60	PRINT*	SHOLES	306
61	PRINT*, 'INCORRECT NUMBER !!!!! TRY AGAIN !!!'	SHOLES	307
62	PRINT*, ' -OR- ENTER "0" TO ESCAPE FROM ',	SHOLES	308
63	+ '"DISPLAY" MODE'	SHOLES	309
64	GOTO 100	SHOLES	310

65 *		SHOLES	311
66 *** ABORT 'DISPLAY' MODE		SHOLES	312
67 ELSE IF (N .EQ. 0) THEN		SHOLES	313
68 PRINT*		SHOLES	314
69 PRINT*, ' "DISPLAY" MODE ABORTED !!!'		SHOLES	315
70 *		SHOLES	316
71 *** DISPLAY LINE OF DATA		SHOLES	317
72 ELSE IF ((N .GT. 0) .AND. (N .LE. HTOT)) THEN		SHOLES	318
73 PRINT*		SHOLES	319
74 CALL DISPLAY(N, COMMAND)		SHOLES	320
75 *		SHOLES	321
76 END IF		SHOLES	322
77 END IF		SHOLES	323
78 END IF		SHOLES	324
79 *		SHOLES	325
80 *-----		SHOLES	326
81 *** INSERT LINE ***		SHOLES	327
82 *-----		SHOLES	328
83 IF (COMMAND .EQ. 2) THEN		SHOLES	329
84 *		SHOLES	330
85 *** INDICATE EMPTY DATA FILE		SHOLES	331
86 IF (HTOT .EQ. 0) THEN		SHOLES	332
87 PRINT*		SHOLES	333
88 PRINT*, 'DATA FILE IS EMPTY !!!'		SHOLES	334
89 *		SHOLES	335
90 *** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE		SHOLES	336
91 ELSE		SHOLES	337
92 200 PRINT*		SHOLES	338
93 PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS '		SHOLES	339
94 + 'TO BE INSERTED'		SHOLES	340
95 PRINT*, ' (ENTER "0" TO ESCAPE "INSERTION" MODE)'		SHOLES	341
96 REWIND 1		SHOLES	342
97 READ(1,*,END=200) N		SHOLES	343
98 *		SHOLES	344
99 *** CHECK FOR VALID LINE NUMBER		SHOLES	345
100 IF ((N .LT. 0) .OR. (N .GT. HTOT)) THEN		SHOLES	346
101 PRINT*		SHOLES	347
102 PRINT*, 'INCORRECT LINE NUMBER !!!'		SHOLES	348
103 PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE',		SHOLES	349
104 + '"INSERTION" MODE'		SHOLES	350
105 GOTO 200		SHOLES	351
106 *		SHOLES	352
107 *** ABORT INSERTION MODE		SHOLES	353
108 ELSE IF (N .EQ. 0) THEN		SHOLES	354
109 PRINT*		SHOLES	355
110 PRINT*, ' "INSERTION" MODE ABORTED'		SHOLES	356
111 *		SHOLES	357
112 *** MAKE ROOM FOR NEW LINE OF DATA		SHOLES	358
113 ELSE IF ((N .GT. 0) .AND. (N .LE. HTOT)) THEN		SHOLES	359
114 DO 230 X = HTOT,N,-1		SHOLES	360
115 DO 210 Y = 1,4		SHOLES	361
116 HOLE(X+1,Y) = HOLE(X,Y)		SHOLES	362
117 210 CONTINUE		SHOLES	363
118 230 CONTINUE		SHOLES	364
119 *		SHOLES	365
120 *** ENTER DATA FOR NEW LINE		SHOLES	366
121 HTOT = HTOT + 1		SHOLES	367
122 CALL DATAIN (1,N)		SHOLES	368
123 *		SHOLES	369
124 PRINT*		SHOLES	370
125 PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N		SHOLES	371
126 CALL DISPLAY(N, COMMAND)		SHOLES	372
127 END IF		SHOLES	373
128 END IF		SHOLES	374

129	END IF	SHOLES	375
130	*	SHOLES	376
131	*-----	SHOLES	377
132	*** DELETE LINE ***	SHOLES	378
133	*-----	SHOLES	379
134	IF (COMMAND .EQ. 3) THEN	SHOLES	380
135	*	SHOLES	381
136	*** INDICATE EMPTY DATA FILE	SHOLES	382
137	IF (HTOT .EQ. 0) THEN	SHOLES	383
138	PRINT*	SHOLES	384
139	PRINT*, 'DATA FILE IS EMPTY !!!'	SHOLES	385
140	*	SHOLES	386
141	*** READ NUMBER OF LINE TO BE DELETED	SHOLES	387
142	ELSE	SHOLES	388
143	300 PRINT*	SHOLES	389
144	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'	SHOLES	390
145	PRINT*, ' (ENTER "0" TO ESCAPE DELETION MODE)'	SHOLES	391
146	REWIND 1	SHOLES	392
147	READ(1,*,END=300) N	SHOLES	393
148	*	SHOLES	394
149	*** CHECK VALIDITY OF LINE NUMBER	SHOLES	395
150	IF ((N .GT. HTOT) .OR. (N .LT. 0)) THEN	SHOLES	396
151	PRINT*	SHOLES	397
152	PRINT*, ' INCORRECT NUMBER !!!'	SHOLES	398
153	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE FROM',	SHOLES	399
154	+ "DELETE" MODE'	SHOLES	400
155	GOTO 300	SHOLES	401
156	*	SHOLES	402
157	*** ABORT 'DELETE' MODE	SHOLES	403
158	ELSE IF (N .EQ. 0) THEN	SHOLES	404
159	PRINT*, ' "DELETE" MODE ABORTED'	SHOLES	405
160	*	SHOLES	406
161	*** DOUBLE CHECK CHOICE OF LINE TO BE DELETED	SHOLES	407
162	ELSE IF ((N .GT. 0) .AND. (N .LE. HTOT)) THEN	SHOLES	408
163	PRINT*	SHOLES	409
164	PRINT*, 'DOUBLE CHECK !!!'	SHOLES	410
165	PRINT*, ' DO YOU WANT TO DELETE THE FOLLOWING LINE?:'	SHOLES	411
166	CALL DISPLAY(N, COMMAND)	SHOLES	412
167	305 PRINT*, ' ENTER (1) YES OR (2) NO'	SHOLES	413
168	REWIND 1	SHOLES	414
169	READ(1,*,END=305) ANSWER	SHOLES	415
170	*	SHOLES	416
171	*** DELETE LINE	SHOLES	417
172	IF (ANSWER .EQ. 1) THEN	SHOLES	418
173	DO 330 X = N, HTOT - 1	SHOLES	419
174	DO 310 Y = 1,4	SHOLES	420
175	HOLE(X,Y) = HOLE(X+1,Y)	SHOLES	421
176	310 CONTINUE	SHOLES	422
177	330 CONTINUE	SHOLES	423
178	HTOT = HTOT - 1	SHOLES	424
179	PRINT*	SHOLES	425
180	PRINT*, 'LINE # ',N,' DELETED'	SHOLES	426
181	END IF	SHOLES	427
182	*	SHOLES	428
183	END IF	SHOLES	429
184	END IF	SHOLES	430
185	END IF	SHOLES	431
186	*	SHOLES	432
187	*-----	SHOLES	433
188	*** DISPLAY ALL DATA ***	SHOLES	434
189	*-----	SHOLES	435
190	IF (COMMAND .EQ. 4) THEN	SHOLES	436
191	*	SHOLES	437
192	*** INDICATE EMPTY DATA FILE	SHOLES	438

193	IF (HTOT .EQ. 0) THEN	SHOLES	439
194	PRINT*	SHOLES	440
195	PRINT*, 'DATA FILE IS EMPTY !!!'	SHOLES	441
196 *		SHOLES	442
197 *** DISPLAY DATA		SHOLES	443
198 ELSE		SHOLES	444
199 PRINT*		SHOLES	445
200 CALL DISPLAY(N, COMMAND)		SHOLES	446
201 *		SHOLES	447
202 END IF		SHOLES	448
203 END IF		SHOLES	449
204 *		SHOLES	450
205 *-----		SHOLES	451
206 *** ADD DATA ***		SHOLES	452
207 *-----		SHOLES	453
208 IF (COMMAND .EQ. 5) THEN		SHOLES	454
209 *		SHOLES	455
210 *** ENTER DATA		SHOLES	456
211 500 HTOT = HTOT + 1		SHOLES	457
212 CALL DATAIN (0,HTOT)		SHOLES	458
213 510 PRINT*		SHOLES	459
214 PRINT*, 'DO YOU WANT TO ENTER MORE DATA? (1) YES (2) NO'		SHOLES	460
215 PRINT*, ' ENTER A NUMBER !!!'		SHOLES	461
216 REWIND 1		SHOLES	462
217 READ(1,*,END=510) ANSWER		SHOLES	463
218 *		SHOLES	464
219 *** CHECK VALIDITY OF NUMBER		SHOLES	465
220 IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN		SHOLES	466
221 GOTO 510		SHOLES	467
222 *		SHOLES	468
223 *** ENTER MORE DATA		SHOLES	469
224 ELSE IF (ANSWER .EQ. 1) THEN		SHOLES	470
225 GOTO 500		SHOLES	471
226 *		SHOLES	472
227 *** DISCONTINUE DATA ENTRY		SHOLES	473
228 ELSE IF (ANSWER .EQ. 2) THEN		SHOLES	474
229 PRINT*		SHOLES	475
230 PRINT*, 'DATA ENTRY DISCONTINUED'		SHOLES	476
231 *		SHOLES	477
232 END IF		SHOLES	478
233 END IF		SHOLES	479
234 *		SHOLES	480
235 *-----		SHOLES	481
236 *** STORE DATA AND PROGRAM ***		SHOLES	482
237 *-----		SHOLES	483
238 IF (COMMAND .EQ. 6) THEN		SHOLES	484
239 600 PRINT*		SHOLES	485
240 PRINT*, 'DOUBLE CHECK !!!'		SHOLES	486
241 PRINT*, ' DO YOU YOU WANT TO STORE THIS DATA AND END PROG'		SHOLES	487
242 PRINT*, ' NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE '		SHOLES	488
243 PRINT*, ' OF THE SAME NAME !!!'		SHOLES	489
244 PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'		SHOLES	490
245 REWIND 1		SHOLES	491
246 READ(1,*,END=600) ANSWER		SHOLES	492
247 *		SHOLES	493
248 *** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM		SHOLES	494
249 IF (ANSWER .EQ. 1) THEN		SHOLES	495
250 QUIT = 1		SHOLES	496
251 RETURN		SHOLES	497
252 *		SHOLES	498
253 *** ABORT 'STORING' MODE		SHOLES	499
254 ELSE IF (ANSWER .EQ. 2) THEN		SHOLES	500
255 PRINT*		SHOLES	501
256 PRINT*, ' "STORING" MODE DISCONTINUED'		SHOLES	502

257 *		SHOLES	503
258 *** CHECK VALIDITY OF ANSWER		SHOLES	504
259 ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN		SHOLES	505
260 GOTO 600		SHOLES	506
261 *		SHOLES	507
262 END IF		SHOLES	508
263 END IF		SHOLES	509
264 *		SHOLES	510
265 *-----		SHOLES	511
266 *** END PROGRAM WITHOUT STORING DATA ***		SHOLES	512
267 *-----		SHOLES	513
268 IF (COMMAND .EQ. 7) THEN		SHOLES	514
269 700 PRINT*		SHOLES	515
270 PRINT*, 'DOUBLE CHECK !!!'		SHOLES	516
271 PRINT*, ' DO YOU WANT TO END THIS PROGRAM ',		SHOLES	517
272 + 'WITHOUT STORING DATA?'		SHOLES	518
273 PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'		SHOLES	519
274 REWIND 1		SHOLES	520
275 READ(1,*,END=700) ANSWER		SHOLES	521
276 *		SHOLES	522
277 *** SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM		SHOLES	523
278 IF (ANSWER .EQ. 1) THEN		SHOLES	524
279 ABORT = 1		SHOLES	525
280 RETURN		SHOLES	526
281 *		SHOLES	527
282 *** ABORT 'STORING' MODE		SHOLES	528
283 ELSE IF (ANSWER .EQ. 2) THEN		SHOLES	529
284 PRINT*		SHOLES	530
285 PRINT*, ' "ABORTION" MODE DISCONTINUED'		SHOLES	531
286 *		SHOLES	532
287 *** CHECK VALIDITY OF ANSWER		SHOLES	533
288 ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN		SHOLES	534
289 GOTO 700		SHOLES	535
290 *		SHOLES	536
291 END IF		SHOLES	537
292 END IF		SHOLES	538
293 *		SHOLES	539
294 *-----		SHOLES	540
295 *** LOOP TO BEGINNING OF 'MANIP' SUBROUTINE		SHOLES	541
296 *-----		SHOLES	542
297 GOTO 10		SHOLES	543
298 *		SHOLES	544
299 RETURN		SHOLES	545
TRIVIAL* NO PATH TO THIS STATEMENT			
300 END		SHOLES	546

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	2	DUMMY-ARG	INTEGER	
ANSWER	1400B		INTEGER	
COMMAND	1405B		INTEGER	
DIR	NONE	UNUSED/*S*	CHAR*3	
DOK	NONE	UNUSED/*S*	INTEGER	
FLAG1	1401B		INTEGER	
FROM	NONE	UNUSED/*S*	CHAR*3	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
INSERT	NONE	UNUSED/*S*	INTEGER	
LOK	NONE	UNUSED/*S*	INTEGER	
N	1402B		INTEGER	
NOK	NONE	UNUSED/*S*	INTEGER	

OK	NONE	UNUSED/*S*	INTEGER
OK1	NONE	UNUSED/*S*	INTEGER
OK2	NONE	UNUSED/*S*	INTEGER
QUIT	1 DUMMY-ARG		INTEGER
TEMP	NONE	UNUSED/*S*	INTEGER
TO	NONE	UNUSED/*S*	CHAR*3
V	NONE	UNUSED/*S*	INTEGER
X	1403B		INTEGER
Y	1404B		INTEGER

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

HMAX	INTEGER	35
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

DATA IN		2	SUBROUTINE
DISPLAY		2	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

10	7B	26	310	INACTIVE	DO-TERM	176
100	50B	52	330	INACTIVE	DO-TERM	177
200	133B	92	500	437B		211
210	INACTIVE	DO-TERM	117	510	443B	213
230	INACTIVE	DO-TERM	118	600	504B	239
300	263B	143	700	556B		269
305	332B	167				

--ENTRY POINTS--(LO=A)

-NAME-----ADDRESS--ARGS---

MANIP	5B	2
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE1	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	1414B = 780
CM LABELLED COMMON LENGTH	54B = 44
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.388 SECONDS

1 TRIVIAL ERROR IN MANIP

1	SUBROUTINE DISPLAY (LINE, COMMAND)	SHOLES	547
2	*****	COMH	1
3	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS	***COMH	2
4	*****	COMH	3
5	INTEGER HMAX	COMH	4
6	PARAMETER (HMAX = 35)	COMH	5
7	COMMON /HOLEN/ HTOT, HERR	COMH	6
8	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
9	INTEGER HTOT, HERR	COMH	8
10	CHARACTER * 3 HOLE	COMH	9
11	* =====	COMH	10
12	* DESCRIPTION OF ARRAYS	COMH	11
13	* =====	COMH	12
14	* ROOM IDENTIFICATION APERTURE ID	COMH	13
15	* -----	COMH	14
16	* DIRECTION FROM ROOM TO ROOM	COMH	15
17	* -----	COMH	16
18	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17
19	* A3 A3 A3 A3	COMH	18
20	*****	COMH	19
21	*****	COMH	20
22	INTEGER LINE, COMMAND, N	SHOLES	549
23	1000 FORMAT (8(3X,A))	SHOLES	550
24	2000 FORMAT (4X,I3,8X,A3,7X,A3,3X,A3,2X,A3)	SHOLES	551
25	PRINT 1000, 'LINE #', 'DIRECTION', 'FROM', 'TO', 'ID'	SHOLES	552
26	IF (COMMAND .EQ. 4) THEN	SHOLES	553
27	DO 10 N = 1,HTOT	SHOLES	554
28	PRINT 2000, N,HOLE(N,1),HOLE(N,2),HOLE(N,3),HOLE(N,4)	SHOLES	555
29	10 CONTINUE	SHOLES	556
30	ELSE	SHOLES	557
31	PRINT 2000, LINE,HOLE(LINE,1),HOLE(LINE,2),HOLE(LINE,3),	SHOLES	558
32	+ HOLE(LINE,4)	SHOLES	559
33	END IF	SHOLES	560
34	RETURN	SHOLES	561
35	END	SHOLES	562

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

COMMAND	2	DUMMY-ARG	INTEGER	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
LINE	1	DUMMY-ARG	INTEGER	
N	204B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)
-NAME---TYPE-----VALUE

HMAX	INTEGER	35
------	---------	----

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

10	INACTIVE	DO-TERM	29
1000	123B	FORMAT	23
2000	125B	FORMAT	24

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

DISPLAY	5B	2
---------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	210B =	136
CM LABELLED COMMON LENGTH	54B =	44
CM STORAGE USED	61000B =	25088
COMPILE TIME	0.068	SECONDS

1	INTEGER FUNCTION VAL(String)	SHOLES	563
2	C** RETURNS THE INTEGER VALUE OF A STRING.	SHOLES	564
3	INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN	SHOLES	565
4	CHARACTER * (*) STRING	SHOLES	566
5	L = GETLEN(String)	SHOLES	567
6	NUMBER = 0	SHOLES	568
7	DO 10 X = L,1,-1	SHOLES	569
8	EXP = L - X	SHOLES	570
9	DIGIT = ICHAR(String(X:X)) - 16	SHOLES	571
10	NUMBER = NUMBER + DIGIT*10**EXP	SHOLES	572
11	10 CONTINUE	SHOLES	573
12	VAL = NUMBER	SHOLES	574
13	RETURN	SHOLES	575
14	END	SHOLES	576

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

DIGIT	76B		INTEGER
EXP	75B		INTEGER
L	74B		INTEGER
NUMBER	72B		INTEGER
STRING	1	DUMMY-ARG	CHAR*(*)
VAL	71B		INTEGER
X	73B		INTEGER

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	11
----	----------	---------	----

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS---ARGS---

VAL	6B	1
-----	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.039 SECONDS

1	INTEGER FUNCTION GETLEN (STRING)	SHOLES	577
2	C	SHOLES	578
3	C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING	SHOLES	579
4	C	SHOLES	580
5	C	SHOLES	581
6	C ARGUMENT DEFINITIONS --	SHOLES	582
7	C READ ARGUMENTS	SHOLES	583
8	C STRING - STRING WHOSE LENGTH IS TO BE DETERMINED	SHOLES	584
9	C	SHOLES	585
10	CHARACTER * (*) STRING	SHOLES	586
11	C	SHOLES	587
12	C FUNCTION PARAMETERS	SHOLES	588
13	CHARACTER * 1 BLANK	SHOLES	589
14	PARAMETER (BLANK = ' ')	SHOLES	590
15	C	SHOLES	591
16	C LOCAL VARIABLES	SHOLES	592
17	INTEGER NEXT	SHOLES	593
18	C	SHOLES	594
19	C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK	SHOLES	595
20	DO 10 NEXT = LEN(STRING),1,-1	SHOLES	596
21	IF (STRING(NEXT : NEXT) .NE. BLANK) THEN	SHOLES	597
22	GETLEN = NEXT	SHOLES	598
23	RETURN	SHOLES	599
24	END IF	SHOLES	600
25	10 CONTINUE	SHOLES	601
26	C	SHOLES	602
27	C ALL CHARACTERS ARE BLANKS	SHOLES	603
28	GETLEN = 0	SHOLES	604
29	C	SHOLES	605
30	RETURN	SHOLES	606
31	END	SHOLES	607

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

GETLEN	63B	INTEGER
NEXT	64B	INTEGER
STRING	1 DUMMY-ARG	CHAR*(*)

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME---TYPE-----VALUE

BLANK	CHAR*1	' '
-------	--------	-----

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

LEN	INTEGER	1	INTRINSIC
-----	---------	---	-----------

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES---DEF

10	INACTIVE	DO-TERM	25
----	----------	---------	----

FTN 5.1+552 83/12/24 09.11.45 PAGE 20
 FUNCTION GETLEN 74/175 OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS---ARGS---

GETLEN	6B	1
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	70B = 56
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.037 SECONDS

Line	Code	Comments	Line
1	SUBROUTINE LHOLE		1
2	*!!!!!!	LHOLE	2
3	*!!!!	!!!LHOLE	3
4	*!!! LOAD THE CONTENTS OF THE "HOLE" FILE INTO THE "HOLE" ARRAY	!!!LHOLE	4
5	*!!!!	!!!LHOLE	5
6	*!!!!!!	!!!LHOLE	6
7	*****	COMH	1
8	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS	***COMH	2
9	*****	COMH	3
10	INTEGER HMAX	COMH	4
11	PARAMETER (HMAX = 35)	COMH	5
12	COMMON /HOLEN/ HTOT, HERR	COMH	6
13	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
14	INTEGER HTOT, HERR	COMH	8
15	CHARACTER * 3 HOLE	COMH	9
16	* =====	COMH	10
17	* DESCRIPTION OF ARRAYS	COMH	11
18	* =====	COMH	12
19	* ROOM IDENTIFICATION APERTURE ID	COMH	13
20	* -----	COMH	14
21	* DIRECTION FROM ROOM TO ROOM	COMH	15
22	* -----	COMH	16
23	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17
24	* A3 A3 A3 A3	COMH	18
25	*****	COMH	19
26	*****	COMH	20
27	*****	COMF	1
28	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
29	*****	COMF	3
30	INTEGER FMAX	COMF	4
31	PARAMETER (FMAX = 50)	COMF	5
32	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
33	\$ FTOT	COMF	7
34	COMMON /INITILC/ BLDG	COMF	8
35	CHARACTER * 5 BLDG	COMF	9
36	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
37	INTEGER QUALITY, FERR, FTOT	COMF	11
38	*****	COMF	12
39	*****	COMF	13
40	INTEGER GETLEN, R, C	LHOLE	9
41	CHARACTER * 7 PFN	LHOLE	10
42	PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'H'	LHOLE	11
43	HERR = 0	LHOLE	12
44	CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',HERR)	LHOLE	13
45	IF (HERR .EQ. 0) THEN	LHOLE	14
46	OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',	LHOLE	15
47	\$ STATUS='OLD', ACCESS='SEQUENTIAL')	LHOLE	16
48	1000 FORMAT (1X,4(1X,A3))	LHOLE	17
49	HTOT = 0	LHOLE	18
50	DO 10 R = 1,HMAX	LHOLE	19
51	READ (3,1000,END=20)(HOLE(R,C),C=1,4)	LHOLE	20
52	HTOT = HTOT + 1	LHOLE	21
53	10 CONTINUE	LHOLE	22
54	20 CONTINUE	LHOLE	23
55	CLOSE(3,STATUS='DELETE')	LHOLE	24
56	ELSE IF (HERR .EQ. 2) THEN	LHOLE	25
57	CALL WARNING (1)	LHOLE	26
58	ELSE	LHOLE	27
59	CALL WARNING (2)	LHOLE	28
60	END IF	LHOLE	29
61	RETURN	LHOLE	30
62	END	LHOLE	31

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	214B		INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
PFN	215B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
R	213B		INTEGER	
RFLAG	3B	/INITILN/	REAL	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
HMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARCS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	53
20	73B		54
1000	130B	FORMAT	48

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARCS---

LHOLE	5B	0
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE3	AUX/FMT/SEQ
-------	-------------

--STATISTICS--

PROGRAM-UNIT LENGTH	222B = 146
CM LABELLED COMMON LENGTH	145B = 101
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.086 SECONDS

1	SUBROUTINE ERROR(IERR)	ERROR	1
2	CHARACTER*45 MESSAGE(20)	ERROR	2
3	DATA MESSAGE(1) //'MATERIALS DATA BASE IS EMPTY	' / ERROR	3
4	DATA MESSAGE(2) //'FREQUENCY IS OUT OF RANGE	' / ERROR	4
5	DATA MESSAGE(3) //'THIS MATERIAL IS NOT IN DATA BASE	' / ERROR	5
6	DATA MESSAGE(4) //'DENOMINATOR IS ZERO	' / ERROR	6
7	DATA MESSAGE(5) //'FILE HANDLING ERROR	' / ERROR	7
8	DATA MESSAGE(6) //'ERROR CODE IS OUT OF RANGE	' / ERROR	8
9	DATA MESSAGE(7) //'ERROR CODE IS OUT OF RANGE	' / ERROR	9
10	DATA MESSAGE(8) //'ERROR CODE IS OUT OF RANGE	' / ERROR	10
11	DATA MESSAGE(9) //'ERROR CODE IS OUT OF RANGE	' / ERROR	11
12	DATA MESSAGE(10) //'ERROR CODE IS OUT OF RANGE	' / ERROR	12
13	DATA MESSAGE(11) //'ERROR CODE IS OUT OF RANGE	' / ERROR	13
14	DATA MESSAGE(12) //'ERROR CODE IS OUT OF RANGE	' / ERROR	14
15	DATA MESSAGE(13) //'ERROR CODE IS OUT OF RANGE	' / ERROR	15
16	DATA MESSAGE(14) //'ERROR CODE IS OUT OF RANGE	' / ERROR	16
17	DATA MESSAGE(15) //'ERROR CODE IS OUT OF RANGE	' / ERROR	17
18	DATA MESSAGE(16) //'ERROR CODE IS OUT OF RANGE	' / ERROR	18
19	DATA MESSAGE(17) //'ERROR CODE IS OUT OF RANGE	' / ERROR	19
20	DATA MESSAGE(18) //'ERROR CODE IS OUT OF RANGE	' / ERROR	20
21	DATA MESSAGE(19) //'ERROR CODE IS OUT OF RANGE	' / ERROR	21
22	DATA MESSAGE(20) //'ERROR CODE IS OUT OF RANGE	' / ERROR	22
23	IERRM=5	ERROR	23
24	IF(IERR.GT.IERRM) IERR=20	ERROR	24
25	WRITE(6,10) IERR,MESSAGE(IERR)	ERROR	25
26	10 FORMAT(' ***ERROR NUMBER = ',I5,' *** ',A45)	ERROR	26
27	CALL PMDSTOP	ERROR	27
28	STOP 'ERROR'	ERROR	28
29	END	ERROR	29

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

IERR	.	1	DUMMY-ARG	INTEGER	
IERRM		210B		INTEGER	
MESSAGE		56B		CHAR*45	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PMDSTOP		0	SUBROUTINE
---------	--	---	------------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	36B	FORMAT	26
----	-----	--------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

ERROR	5B	1
-------	----	---

FTN 5.1+552 83/12/24. 09.11.45 PAGE 24
SUBROUTINE ERROR 74/175 OPT=0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	213B = 139
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.055 SECONDS

1	SUBROUTINE WARNING(ERR)	WARNING	1
2	INTEGER ERR, ERRM	WARNING	2
3	CHARACTER*45 MESSAGE(20)	WARNING	3
4	DATA MESSAGE(1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	4
5	DATA MESSAGE(2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE	WARNING	5
6	DATA MESSAGE(3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG	WARNING	6
7	DATA MESSAGE(4)/'FILE HANDLING PROBLEM ON "MATTER" FILE	WARNING	7
8	DATA MESSAGE(5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	8
9	DATA MESSAGE(6)/'FILE HANDLING PROBLEM ON "TYPE" FILE	WARNING	9
10	DATA MESSAGE(7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	10
11	DATA MESSAGE(8)/'FILE HANDLING PROBLEM ON "WALL" FILE	WARNING	11
12	DATA MESSAGE(9)/'HEIGHT AND WIDTH OF ROOM MISSING	WARNING	12
13	DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING	WARNING	13
14	DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG	WARNING	14
15	DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE	WARNING	15
16	DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE	WARNING	16
17	DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE	WARNING	17
18	DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE	WARNING	18
19	DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE	WARNING	19
20	DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE	WARNING	20
21	DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE	WARNING	21
22	DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE	WARNING	22
23	DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE	WARNING	23
24	ERRM=12	WARNING	24
25	IERR = ERR	WARNING	25
26	IF(ERR.GT.ERRM) IERR=20	WARNING	26
27	WRITE(6,20)	WARNING	27
28	WRITE(6,10) ERR,MESSAGE(IERR)	WARNING	28
29	WRITE(6,20)	WARNING	29
30 10	FORMAT(' ***WARNING NUMBER = ',I5,' *** ',A45)	WARNING	30
31 20	FORMAT(' ')	WARNING	31
32	RETURN	WARNING	32
33	END	WARNING	33

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ERR	1	DUMMY-ARG	INTEGER	
ERRM	60B		INTEGER	
IERR	213B		INTEGER	
MESSAGE	61B		CHAR*45	20

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

10	34B	FORMAT	30
20	42B	FORMAT	31

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

WARNING	5B	1
---------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	216B = 142
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.057 SECONDS

Appendix 9.4 Listing of Computer Program SWALLS

```

1      PROGRAM SWALLS (INPUT,TAPE1=INPUT)                                SWALLS      1
2      *                                                                    SWALLS      2
3      *THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH WALL    SWALLS      3
4      *IN THE BUILDING AND STORES IT. THE FILE NAME IS CREATED BY        SWALLS      4
5      *ATTACHING "B" TO THE FRONT OF AND "W" TO THE BACK OF THE BUILDING  SWALLS      5
6      *IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE        SWALLS      6
7      *THAN 5 ALPHANUMERIC CHARACTERS.                                    SWALLS      7
8                                                                    SWALLS      8
9      *****COMF 1
10     *** COMMON FOR INITIAL PARAMETERS                                ***COMF 2
11     *****COMF 3
12     INTEGER FMAX                                                    COMF 4
13     PARAMETER (FMAX = 50)                                           COMF 5
14     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF 6
15     $ FTOT                                                           COMF 7
16     COMMON /INITILC/ BLDG                                           COMF 8
17     CHARACTER * 5 BLDG                                              COMF 9
18     REAL FREQ, AFLAG, RFLAG, FREQA                                  COMF 10
19     INTEGER QUALITY, FERR, FTOT                                     COMF 11
20     *****COMF 12
21     *****COMF 13
22     *****COMR 1
23     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS        ***COMR 2
24     *****COMR 3
25     INTEGER RMAX                                                    COMR 4
26     PARAMETER (RMAX = 20)                                           COMR 5
27     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)    COMR 6
28     INTEGER NROOMS                                                  COMR 7
29     REAL ROOM                                                       COMR 8
30     *****COMR 9
31     *****COMR 10
32     *****COMW 1
33     *** COMMON FOR DATABASE OF WALL PARAMETERS                        ***COMW 2
34     *****COMW 3
35     INTEGER WMAX                                                    COMW 4
36     PARAMETER (WMAX = 75)                                           COMW 5
37     COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                         COMW 6
38     COMMON /WALLC/ WALL(WMAX,4)                                     COMW 7
39     INTEGER WTOT,WERR                                              COMW 8
40     REAL WDIM                                                       COMW 9
41     CHARACTER *3 WALL                                              COMW 10
42     * ===== COMW 11
43     * DESCRIPTION OF ARRAYS COMW 12
44     * ===== COMW 13
45     * WALL IDENTIFICATION COMW 14
46     * ----- COMW 15
47     * DIRECTION FROM TO COMW 16
48     * ROOM ROOM COMW 17
49     * ----- COMW 18
50     * WALL(X,1) WALL(X,2) WALL(X,3) COMW 19
51     * A3 A3 A3 COMW 20
52     * ===== COMW 21
53     * WALL PARAMETERS COMW 22
54     * ----- COMW 23
55     * MATERIAL HEIGHT WIDTH LAYER THICKNESS COMW 24
56     * ----- COMW 25
57     * WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3) COMW 26
58     * A3 F8.2 F8.2 F8.2 COMW 27
59     *****COMW 28
60     *****COMW 29
61     INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE          SWALLS 12
62     INTEGER IERR                                                    SWALLS 13
63     CHARACTER * 7 PFN                                              SWALLS 14
64     *                                                                SWALLS 15

```

65	*	INITIALIZATION	SWALLS	16
66		QUIT = 0	SWALLS	17
67		WTOT = 0	SWALLS	18
68		ABORT = 0	SWALLS	19
69	100	PRINT*	SWALLS	20
70		PRINT *, 'ENTER BUILDING IDENTIFICATION (E.G. '101')'	SWALLS	21
71		PRINT *, ' (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'	SWALLS	22
72		REWIND 1	SWALLS	23
73		READ(1,*,END=100) BLDG	SWALLS	24
74			SWALLS	25
75		IF (GETLEN(BLDG) .GT. 5) THEN	SWALLS	26
76		GO TO 100	SWALLS	27
77		END IF	SWALLS	28
78		PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'W'	SWALLS	29
79	*		SWALLS	30
80	***	LOAD DATA FROM EXISTING FILE IF NECESSARY	SWALLS	31
81	200	PRINT*	SWALLS	32
82		PRINT*, 'WILL THIS BE'	SWALLS	33
83		PRINT*, ' (1) A MODIFICATION OF AN EXISTING FILE?'	SWALLS	34
84		PRINT*, ' (2) A NEW FILE?'	SWALLS	35
85		PRINT*, 'ENTER A NUMBER !!!!'	SWALLS	36
86		REWIND 1	SWALLS	37
87		READ(1,*,END=200) OLDFILE	SWALLS	38
88		IF ((OLDFILE .NE. 1) .AND. (OLDFILE .NE. 2)) THEN	SWALLS	39
89		GOTO 200	SWALLS	40
90		ELSE IF (OLDFILE .EQ. 1) THEN	SWALLS	41
91	*		SWALLS	42
92	***	CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	SWALLS	43
93		IERR = 0	SWALLS	44
94		CALL PF ('GET', 0, PFN(1:GETLEN(PFN)), 'RC', IERR)	SWALLS	45
95		IF (IERR .EQ. 2) THEN	SWALLS	46
96		PRINT*	SWALLS	47
97		PRINT *, 'FILE ', PFN, ' NOT FOUND'	SWALLS	48
98		PRINT*, 'PROGRAM ABORTED!!!'	SWALLS	49
99		PRINT*	SWALLS	50
100		PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART ',	SWALLS	51
101	+	'PROGRAM'	SWALLS	52
102		PRINT*	SWALLS	53
103		STOP	SWALLS	54
104	*		SWALLS	55
105		ELSE	SWALLS	56
106		CALL LWALL	SWALLS	57
107		IF (WERR .NE. 0) CALL ERROR(5)	SWALLS	58
108		ENDIF	SWALLS	59
109		ELSE IF (OLDFILE .EQ. 2) THEN	SWALLS	60
110	*		SWALLS	61
111	***	CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	SWALLS	62
112		IERR = 0	SWALLS	63
113		CALL PF ('GET', 0, PFN(1:GETLEN(PFN)), 'RC', IERR)	SWALLS	64
114		IF (IERR .EQ. 0) THEN	SWALLS	65
115		PRINT*	SWALLS	66
116		PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ', BLDG	SWALLS	67
117		PRINT*	SWALLS	68
118		PRINT*, 'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',	SWALLS	69
119	+	'OVER THE OLD FILE.'	SWALLS	70
120	250	PRINT*	SWALLS	71
121		PRINT*, 'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'	SWALLS	72
122		PRINT*, 'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'	SWALLS	73
123		REWIND 1	SWALLS	74
124		READ(1,*,END=250) ANSWER	SWALLS	75
125		PRINT*	SWALLS	76
126		PRINT*, 'PROGRAM HAS BEEN ABORTED, PER YOUR REQUEST'	SWALLS	77
127		PRINT*	SWALLS	78
128		IF (ANSWER .EQ. 1) THEN	SWALLS	79

129	STOP	SWALLS	80
130	ELSE IF (ANSWER .EQ. 2) THEN	SWALLS	81
131 9090	CONTINUE	SWALLS	82
132	ELSE	SWALLS	83
133	GOTO 250	SWALLS	84
134	ENDIF	SWALLS	85
135	ELSE IF (IERR .EQ. 2) THEN	SWALLS	86
136 *		SWALLS	87
137 *	NO DATA FILE EXISTS FOR THIS BUILDING AND DATA ENTRY	SWALLS	88
138 *	CAN CONTINUE	SWALLS	89
139 *		SWALLS	90
140 9091	CONTINUE	SWALLS	91
141	ELSE	SWALLS	92
142 *		SWALLS	93
143 *	**PERMANENT FILE ERROR	SWALLS	94
144 *		SWALLS	95
145	PRINT*	SWALLS	96
146	PRINT*, 'PROGRAM ABORTED !!!'	SWALLS	97
147	PRINT*, ' SOME PERMANENT FILE ERROR HAS OCCURRED'	SWALLS	98
148	PRINT*, ' DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',	SWALLS	99
149	+ 'AND TRY AGAIN'	SWALLS	100
150	STOP	SWALLS	101
151	ENDIF	SWALLS	102
152	PRINT*	SWALLS	103
153	PRINT*, ' BEGIN ENTERING DATA'	SWALLS	104
154	*****	SWALLS	105
155 *	CHECK TO SEE IF THERE IS ENOUGH ARRAY SPACE	SWALLS	106
156	*****	SWALLS	107
157 300	IF (WTOT.GE.WMAX) THEN	SWALLS	108
158	WTOT = WMAX	SWALLS	109
159	PRINT *, 'DATA ENTRY ABORTED.'	SWALLS	110
160	PRINT *, 'MAXIMUM NUMBER OF DATA LINES IN FILE WOULD'	SWALLS	111
161	PRINT *, ' HAVE BEEN EXCEEDED. NO MORE THAN ',WMAX	SWALLS	112
162	PRINT *, ' DATA LINES ARE ALLOWED.'	SWALLS	113
163	PRINT *, ' TO INCREASE THE MAXIMUM NUMBER OF ENTRIES ALLOWED,'	SWALLS	114
164	PRINT *, ' CHANGE THE PARAMETER "WMAX" IN EACH COMMON OF'	SWALLS	115
165	PRINT *, ' EVERY SUBROUTINE (THERE ARE FOUR PLACES).'	SWALLS	116
166	PRINT *, ' THEN RECOMPILE THE PROGRAM.'	SWALLS	117
167	GOTO 450	SWALLS	118
168	ENDIF	SWALLS	119
169	*****	SWALLS	120
170	WTOT = WTOT + 1	SWALLS	121
171	IF (WTOT .EQ. 1) THEN	SWALLS	122
172	CALL DATAIN(1,WTOT)	SWALLS	123
173	ELSE	SWALLS	124
174	CALL DATAIN (0,WTOT)	SWALLS	125
175	END IF	SWALLS	126
176 400	PRINT*	SWALLS	127
177	PRINT*, 'DO YOU WANT TO ENTER MORE DATA?'	SWALLS	128
178	+ '(1) YES (2) NO'	SWALLS	129
179	PRINT*, ' ENTER A NUMBER !!!'	SWALLS	130
180	REWIND 1	SWALLS	131
181	READ(1,*,END=400) ANSWER	SWALLS	132
182	IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SWALLS	133
183	GOTO 400	SWALLS	134
184	ELSE IF (ANSWER .EQ. 1) THEN	SWALLS	135
185	GOTO 300	SWALLS	136
186	ELSE IF (ANSWER .EQ. 2) THEN	SWALLS	137
187	PRINT*	SWALLS	138
188	PRINT*, 'DATA ENTRY DISCONTINUED'	SWALLS	139
189	END IF	SWALLS	140
190	END IF	SWALLS	141
191 *		SWALLS	142
192	*** MANIPULATE DATA	SWALLS	143

193 450	CALL MANIP (QUIT,ABORT)	SWALLS	144
194 *		SWALLS	145
195 ***	TERMINATE PROGRAM, STORING DATA IF NECESSARY	SWALLS	146
196	IF (QUIT .EQ. 1) THEN	SWALLS	147
197	OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',	SWALLS	148
198	+ ACCESS='SEQUENTIAL',STATUS='NEW')	SWALLS	149
199 500	FORMAT (1X,4(1X,A3),3(1X,F8.2))	SWALLS	150
200	DO 600 N = 1,WTOT	SWALLS	151
201	WRITE (6,500)(WALL(N,Y1), Y1=1,4),(WDIM(N,Y2), Y2=1,3)	SWALLS	152
202 600	CONTINUE	SWALLS	153
203	ENDFILE(6)	SWALLS	154
204	CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))	SWALLS	155
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT			
205	CLOSE(6,STATUS='DELETE')	SWALLS	156
206	PRINT*	SWALLS	157
207	PRINT*, 'DATA HAS BEEN STORED AND PROGRAM TERMINATED'	SWALLS	158
208	END IF	SWALLS	159
209	IF(ABORT .EQ. 1) THEN	SWALLS	160
210	PRINT*	SWALLS	161
211	PRINT*, 'PROGRAM HAS BEEN ABORTED'	SWALLS	162
212	PRINT*, ' NO DATA HAS BEEN STORED !!!'	SWALLS	163
213	END IF	SWALLS	164
214	STOP	SWALLS	165
215	END	SWALLS	166

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	1216B		INTEGER	
AFLAG	2B	/INITILN/	REAL	
ANSWER	1217B		INTEGER	
BLDG	0B	/INITILC/	CHAR*5	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
IERR	1224B		INTEGER	
LINE	NONE	UNUSED/*S*	INTEGER	
N	1221B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
OLDFILE	1220B		INTEGER	
PFN	1225B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
QUIT	1215B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	
Y1	1222B		INTEGER	
Y2	1223B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
RMAX	INTEGER	20
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----	-TYPE-----	-ARGS-----	-CLASS-----	-NAME-----	-TYPE-----	-ARGS-----	-CLASS-----
DATAIN		2	SUBROUTINE	LWALL		0	SUBROUTINE
ERROR		1	SUBROUTINE	MANIP		2	SUBROUTINE
GETLEN	INTEGER	1	FUNCTION	PF		5	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----	-PROPERTIES----	-DEF	-LABEL-ADDRESS-----	-PROPERTIES----	-DEF
100	21B	69	450	344B	193
200	47B	81	500	714B	199
250	166B	120	600	INACTIVE	202
300	246B	157	9090	*NO REFS*	131
400	306B	176	9091	*NO REFS*	140

--ENTRY POINTS--(LO=A)

-NAME---	-ADDRESS---	-ARGS---
SWALLS	14B	0

--I/O UNITS--(LO=A)

-NAME---	-PROPERTIES-----
TAPE1	FMT/SEQ
TAPE6	AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	1231B = 665
CM LABELLED COMMON LENGTH	2057B = 1071
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.317 SECONDS

1 WARNING ERROR IN SWALLS

```

1      SUBROUTINE DATAIN (INSERT,LINE)                                SWALLS    167
2      *****COMR 1
3      *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR 2
4      *****COMR 3
5      INTEGER RMAX                                                    COMR 4
6      PARAMETER (RMAX = 20)                                           COMR 5
7      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)   COMR 6
8      INTEGER NROOMS                                                  COMR 7
9      REAL ROOM                                                       COMR 8
10     *****COMR 9
11     *****COMR 10
12     *****COMW 1
13     *** COMMON FOR DATABASE OF WALL PARAMETERS ***COMW 2
14     *****COMW 3
15     INTEGER WMAX                                                    COMW 4
16     PARAMETER (WMAX = 75)                                           COMW 5
17     COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                        COMW 6
18     COMMON /WALLC/ WALL(WMAX,4)                                     COMW 7
19     INTEGER WTOT,WERR                                               COMW 8
20     REAL WDIM                                                       COMW 9
21     CHARACTER *3 WALL                                              COMW 10
22     * ===== COMW 11
23     ** DESCRIPTION OF ARRAYS COMW 12
24     * ===== COMW 13
25     * WALL IDENTIFICATION COMW 14
26     * ----- COMW 15
27     * DIRECTION FROM TO COMW 16
28     * ROOM ROOM COMW 17
29     * ----- COMW 18
30     * WALL(X,1) WALL(X,2) WALL(X,3) COMW 19
31     * A3 A3 A3 COMW 20
32     * ===== COMW 21
33     * WALL PARAMETERS COMW 22
34     * ----- COMW 23
35     * MATERIAL HEIGHT WIDTH LAYER THICKNESS COMW 24
36     * ----- COMW 25
37     * WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3) COMW 26
38     * A3 F8.2 F8.2 F8.2 COMW 27
39     *****COMW 28
40     *****COMW 29
41     INTEGER ANSWER,LOK,DOK,NOK,GETLEN,VAL, INSERT,LINE,V           SWALLS    170
42     CHARACTER *3 DIR,FROM,TO,MAT SWALLS 171
43     99 IF ( INSERT .EQ. 1 ) THEN SWALLS 172
44         ANSWER = 1 SWALLS 173
45         INSERT = 1 SWALLS 174
46     ELSE SWALLS 175
47 100 PRINT* SWALLS 176
48     PRINT*, 'IS THIS THE FIRST LAYER OF A WALL (1) YES (2) NO' SWALLS 177
49     PRINT*, ' ENTER "0" TO ESCAPE "DATA ENTRY" MODE' SWALLS 178
50     PRINT*, ' ENTER A NUMBER!!' SWALLS 179
51     REWIND 1 SWALLS 180
52     READ(1,*,END=100) ANSWER SWALLS 181
53     END IF SWALLS 182
54     * ..... SWALLS 183
55     IF (ANSWER .EQ. 0) THEN SWALLS 184
56         WTOT = WTOT - 1 SWALLS 185
57         PRINT* SWALLS 186
58         PRINT*, 'DATA ENTRY MODE ABORTED' SWALLS 187
59     END IF SWALLS 188
60     IF ((ANSWER .NE. 2) SWALLS 189
61     + .AND. (ANSWER .NE. 1) SWALLS 190
62     + .AND. (ANSWER .NE. 0)) THEN SWALLS 191
63         PRINT* SWALLS 192
64         PRINT*, 'INCORRECT NUMBER!!' SWALLS 193

```

65	PRINT*, ' TRY AGAIN!! -OR- ENTER "0" TO ESCAPE DATA ENTRY MODE'	SWALLS	194
66	GOTO 99	SWALLS	195
67	END IF	SWALLS	196
68	IF (ANSWER .EQ. 1) THEN	SWALLS	197
69 200	PRINT*	SWALLS	198
70	PRINT*, 'ENTER DIRECTION (E. G. 'LR')'	SWALLS	199
71	REWIND 1	SWALLS	200
72	READ(1,*,END=200) DIR	SWALLS	201
73	IF ((DIR .NE. 'LR')	SWALLS	202
74	+ .AND. (DIR .NE. 'FB')	SWALLS	203
75	+ .AND. (DIR .NE. 'UD')) THEN	SWALLS	204
76	PRINT*, 'DIRECTION MUST BE 'LR' OR 'FB' OR 'UD''	SWALLS	205
77	PRINT*, 'TRY AGAIN!!!'	SWALLS	206
78	GOTO 200	SWALLS	207
79	END IF	SWALLS	208
80	WALL(LINE,1) = DIR	SWALLS	209
81 *	SWALLS	210
82 300	PRINT*	SWALLS	211
83	PRINT*, 'ENTER "FROM" (E.G. '02' OR 'D1')'	SWALLS	212
84	REWIND 1	SWALLS	213
85	READ(1,*,END=300) FROM	SWALLS	214
86	LOK = 0	SWALLS	215
87	DOK = 0	SWALLS	216
88	NOK = 0	SWALLS	217
89	IF (GETLEN(FROM) .EQ. 2) THEN	SWALLS	218
90	LOK = 1	SWALLS	219
91	END IF	SWALLS	220
92	IF (FROM(1:1) .EQ. 'D') THEN	SWALLS	221
93	V = VAL(FROM(2:2))	SWALLS	222
94	IF ((V .GE. 1) .AND. (V .LE. 6)) THEN	SWALLS	223
95	DOK = 1	SWALLS	224
96	END IF	SWALLS	225
97	END IF	SWALLS	226
98	IF ((ICCHAR(FROM(1:1)) .GE. 16)	SWALLS	227
99	+ .AND. (ICCHAR(FROM(1:1)) .LE. 25)	SWALLS	228
100	+ .AND. (ICCHAR(FROM(2:2)) .GE. 16)	SWALLS	229
101	+ .AND. (ICCHAR(FROM(2:2)) .LE. 25)	SWALLS	230
102	+ .AND. (GETLEN(FROM) .EQ. 2)) THEN	SWALLS	231
103	V = VAL(FROM)	SWALLS	232
104	IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN	SWALLS	233
105	NOK = 1	SWALLS	234
106	END IF	SWALLS	235
107	END IF	SWALLS	236
108	IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1) .OR. (NOK .EQ. 1))) THEN	SWALLS	237
109	WALL(LINE,2) = FROM	SWALLS	238
110	ELSE	SWALLS	239
111	PRINT*	SWALLS	240
112	PRINT*, 'INCORRECT ENTRY. TRY AGAIN!!!'	SWALLS	241
113	GOTO 300	SWALLS	242
114	END IF	SWALLS	243
115 *	SWALLS	244
116 400	PRINT*	SWALLS	245
117	PRINT*, 'ENTER "TO" (E.G. '02' OR 'D1')'	SWALLS	246
118	REWIND 1	SWALLS	247
119	READ(1,*,END=400) TO	SWALLS	248
120	LOK = 0	SWALLS	249
121	DOK = 0	SWALLS	250
122	NOK = 0	SWALLS	251
123	IF (GETLEN(TO) .EQ. 2) THEN	SWALLS	252
124	LOK = 1	SWALLS	253
125	END IF	SWALLS	254
126	IF (TO(1:1) .EQ. 'D') THEN	SWALLS	255
127	V = VAL(TO(2:2))	SWALLS	256
128	IF ((V .GE. 1) .AND. (V .LE. 6)) THEN	SWALLS	257

129	DOK = 1	SWALLS	258
130	END IF	SWALLS	259
131	ENDIF	SWALLS	260
132	IF ((ICAR(TO(1:1)) .GE. 16)	SWALLS	261
133	+ .AND. (ICAR(TO(1:1)) .LE. 25)	SWALLS	262
134	+ .AND. (ICAR(TO(2:2)) .GE. 16)	SWALLS	263
135	+ .AND. (ICAR(TO(2:2)) .LE. 25)	SWALLS	264
136	+ .AND. (GETLEN(TO) .EQ. 2)) THEN	SWALLS	265
137	V = VAL (TO)	SWALLS	266
138	IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN	SWALLS	267
139	NOK = 1	SWALLS	268
140	END IF	SWALLS	269
141	ENDIF	SWALLS	270
142	IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1) .OR. (NOK .EQ. 1))) THEN	SWALLS	271
143	WALL (LINE,3) = TO	SWALLS	272
144	ELSE	SWALLS	273
145	PRINT*	SWALLS	274
146	PRINT*, 'INCORRECT ENTRY. TRY AGAIN!!'	SWALLS	275
147	GOTO 400	SWALLS	276
148	END IF	SWALLS	277
149	IF (FROM .EQ. TO) THEN	SWALLS	278
150	PRINT*	SWALLS	279
151	PRINT*, 'INCORRECT ENTRY!!'	SWALLS	280
152	PRINT*, '"FROM" CANNOT EQUAL "TO"'	SWALLS	281
153	PRINT*, 'CHECK YOUR DATA AND REENTER "FROM" AND "TO"'	SWALLS	282
154	PRINT*	SWALLS	283
155	GOTO 300	SWALLS	284
156	END IF	SWALLS	285
157	IF ((FROM(1:1) .EQ. 'D') .AND. (TO(1:1) .EQ. 'D')) THEN	SWALLS	286
158	PRINT*	SWALLS	287
159	PRINT*, 'INCORRECT ENTRY!!'	SWALLS	288
160	PRINT*, '"FROM" AND "TO" CANNOT BOTH CONTAIN "D"'	SWALLS	289
161	PRINT*, ' CHECK YOUR DATA AND REENTER "FROM" AND "TO"'	SWALLS	290
162	PRINT*	SWALLS	291
163	GOTO 300	SWALLS	292
164	END IF	SWALLS	293
165	*	SWALLS	294
166	440 PRINT*	SWALLS	295
167	PRINT*, 'ENTER HEIGHT, METERS'	SWALLS	296
168	REWIND 1	SWALLS	297
169	READ(1,*,END=440) WDIM(LINE,1)	SWALLS	298
170	*	SWALLS	299
171	460 PRINT*	SWALLS	300
172	PRINT*, 'ENTER WIDTH, METERS'	SWALLS	301
173	REWIND 1	SWALLS	302
174	READ(1,*,END=460) WDIM(LINE,2)	SWALLS	303
175	*	SWALLS	304
176	480 PRINT*	SWALLS	305
177	PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'	SWALLS	306
178	REWIND 1	SWALLS	307
179	READ(1,*,END=480) WDIM(LINE,3)	SWALLS	308
180	*	SWALLS	309
181	500 PRINT*	SWALLS	310
182	PRINT*, 'ENTER "MATERIAL ID" (E.G. 'M01')'	SWALLS	311
183	REWIND 1	SWALLS	312
184	READ(1,*,END=500) MAT	SWALLS	313
185	IF ((GETLEN(MAT).EQ. 3)	SWALLS	314
186	+ .AND. (MAT(1:1) .EQ. 'M')	SWALLS	315
187	+ .AND. (ICAR(MAT(2:2)) .GE. 16)	SWALLS	316
188	+ .AND. (ICAR(MAT(2:2)) .LE. 25)	SWALLS	317
189	+ .AND. (ICAR(MAT(3:3)) .GE. 16)	SWALLS	318
190	+ .AND. (ICAR(MAT(3:3)) .LE. 25)) THEN	SWALLS	319
191	WALL(LINE,4) = MAT	SWALLS	320
192	ELSE	SWALLS	321

193	PRINT*	SWALLS	322
194	PRINT*, 'INCORRECT ENTRY!! TRY AGAIN'	SWALLS	323
195	GOTO 500	SWALLS	324
196	END IF	SWALLS	325
197	END IF	SWALLS	326
198 *	SWALLS	327
199	IF (ANSWER .EQ. 2) THEN	SWALLS	328
200 580	PRINT*	SWALLS	329
201	PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'	SWALLS	330
202	REWIND 1	SWALLS	331
203	READ(1,*,END=580) WDIM(LINE,3)	SWALLS	332
204 *	SWALLS	333
205 600	PRINT*	SWALLS	334
206	PRINT*, 'ENTER "MATERIAL ID" (E.G. 'M01')'	SWALLS	335
207	REWIND 1	SWALLS	336
208	READ(1,*,END=600) MAT	SWALLS	337
209	IF ((GETLEN(MAT) .EQ. 3)	SWALLS	338
210	AND. (MAT(1:1) .EQ. 'M')	SWALLS	339
211	AND. (ICHAR(MAT(2:2)) .GE. 16)	SWALLS	340
212	AND. (ICHAR(MAT(2:2)) .LE. 25)	SWALLS	341
213	AND. (ICHAR(MAT(3:3)) .GE. 16)	SWALLS	342
214	AND. (ICHAR(MAT(3:3)) .LE. 25)) THEN	SWALLS	343
215	WALL(LINE,4) = MAT	SWALLS	344
216	ELSE	SWALLS	345
217	PRINT*	SWALLS	346
218	PRINT*, 'INCORRECT ENTRY!! TRY AGAIN'	SWALLS	347
219	GOTO 600	SWALLS	348
220	END IF	SWALLS	349
221	WALL(LINE,3) = WALL(LINE-1,3)	SWALLS	350
222	WALL(LINE,2) = WALL(LINE-1,2)	SWALLS	351
223	WALL(LINE,1) = WALL(LINE-1,1)	SWALLS	352
224	WDIM(LINE,1) = WDIM(LINE-1,1)	SWALLS	353
225	WDIM(LINE,2) = WDIM(LINE-1,2)	SWALLS	354
226	END IF	SWALLS	355
227	RETURN	SWALLS	356
228	END	SWALLS	357

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ANSWER	1434B		INTEGER	
DIR	1441B		CHAR*3	
DOK	1436B		INTEGER	
FROM	1442B		CHAR*3	
INSERT	1	DUMMY-ARG	INTEGER	
LINE	2	DUMMY-ARG	INTEGER	
LOK	1435B		INTEGER	
MAT	1444B		CHAR*3	
NOK	1437B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
ROOM	0B	/ROOMN/	REAL	676
TO	1443B		CHAR*3	
V	1440B		INTEGER	
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

RMAX	INTEGER	20
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC
VAL	INTEGER	1	FUNCTION

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

99	7B	43	460	477B	171
100	16B	47	480	513B	176
200	70B	69	500	527B	181
300	127B	82	580	615B	200
400	264B	116	600	631B	205
440	463B	166			

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

DATAIN	5B	2
--------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE1	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	1447B = 807
CM LABELLED COMMON LENGTH	1766B = 1014
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.436 SECONDS

1	SUBROUTINE MANIP (QUIT,ABORT)	SWALLS	358
2	*****	COMW	1
3	*** COMMON FOR DATABASE OF WALL PARAMETERS	***COMW	2
4	*****	COMW	3
5	INTEGER WMAX	COMW	4
6	PARAMETER (WMAX = 75)	COMW	5
7	COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR	COMW	6
8	COMMON /WALLC/ WALL(WMAX,4)	COMW	7
9	INTEGER WTOT,WERR	COMW	8
10	REAL WDIM	COMW	9
11	CHARACTER *3 WALL	COMW	10
12	* =====	COMW	11
13	** DESCRIPTION OF ARRAYS	COMW	12
14	* =====	COMW	13
15	* WALL IDENTIFICATION	COMW	14
16	* -----	COMW	15
17	* DIRECTION FROM TO	COMW	16
18	* ROOM ROOM	COMW	17
19	* -----	COMW	18
20	* WALL(X,1) WALL(X,2) WALL(X,3)	COMW	19
21	* A3 A3 A3	COMW	20
22	* =====	COMW	21
23	* WALL PARAMETERS	COMW	22
24	* -----	COMW	23
25	* MATERIAL HEIGHT WIDTH LAYER THICKNESS	COMW	24
26	* -----	COMW	25
27	* WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3)	COMW	26
28	* A3 F8.2 F8.2 F8.2	COMW	27
29	*****	COMW	28
30	*****	COMW	29
31	INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT	SWALLS	360
32	INTEGER TEMP,V,X,Y,COMMAND	SWALLS	361
33	CHARACTER * 3 DIR, FROM, TO, MAT	SWALLS	362
34	*	SWALLS	363
35	10 FLAG1 = 0	SWALLS	364
36	PRINT*	SWALLS	365
37	PRINT*, 'CHOOSE'	SWALLS	366
38	PRINT*, ' (1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES'	SWALLS	367
39	PRINT*, ' (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'	SWALLS	368
40	PRINT*, ' (3) DELETE LINE (6) STORE DATA AND EXIT ',	SWALLS	369
41	+ 'PROGRAM'	SWALLS	370
42	PRINT*, ' (7) EXIT PROGRAM WITHOUT ',	SWALLS	371
43	+ 'STORING DATA'	SWALLS	372
44	PRINT*, 'ENTER A NUMBER !!!'	SWALLS	373
45	PRINT*	SWALLS	374
46	REWIND 1	SWALLS	375
47	READ(1,*,END=10) COMMAND	SWALLS	376
48	*	SWALLS	377
49	*-----	SWALLS	378
50	*** DISPLAY LINE ***	SWALLS	379
51	*-----	SWALLS	380
52	IF (COMMAND .EQ. 1) THEN	SWALLS	381
53	*	SWALLS	382
54	*** INDICATE EMPTY DATA FILE	SWALLS	383
55	IF (WTOT .EQ. 0) THEN	SWALLS	384
56	PRINT*	SWALLS	385
57	PRINT*, 'DATA FILE IS EMPTY !!!'	SWALLS	386
58	*	SWALLS	387
59	*** ENTER NUMBER OF LINE TO BE DISPLAYED	SWALLS	388
60	ELSE	SWALLS	389
61	100 PRINT*	SWALLS	390
62	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'	SWALLS	391
63	PRINT*, ' (ENTER "0" TO ESCAPE DISPLAY MODE) '	SWALLS	392
64	REWIND 1	SWALLS	393

65	READ(1,*,END=100) N	SWALLS	394
66	*	SWALLS	395
67	*** CHECK VALIDITY OF LINE NUMBER	SWALLS	396
68	IF ((N .GT. WTOT) .OR. (N .LT. 0)) THEN	SWALLS	397
69	PRINT*	SWALLS	398
70	PRINT*, 'INCORRECT NUMBER !!!!! TRY AGAIN !!!'	SWALLS	399
71	PRINT*, ' -OR- ENTER "0" TO ESCAPE FROM ',	SWALLS	400
72	+ ' "DISPLAY" MODE'	SWALLS	401
73	GOTO 100	SWALLS	402
74	*	SWALLS	403
75	*** ABORT 'DISPLAY' MODE	SWALLS	404
76	ELSE IF (N .EQ. 0) THEN	SWALLS	405
77	PRINT*	SWALLS	406
78	PRINT*, ' "DISPLAY" MODE ABORTED !!!'	SWALLS	407
79	*	SWALLS	408
80	*** DISPLAY LINE OF DATA	SWALLS	409
81	ELSE IF ((N .GT. 0) .AND. (N .LE. WTOT)) THEN	SWALLS	410
82	PRINT*	SWALLS	411
83	CALL DISPLAY(N, COMMAND)	SWALLS	412
84	*	SWALLS	413
85	END IF	SWALLS	414
86	END IF	SWALLS	415
87	END IF	SWALLS	416
88	*	SWALLS	417
89	*-----	SWALLS	418
90	*** INSERT LINE ***	SWALLS	419
91	*-----	SWALLS	420
92	IF (COMMAND .EQ. 2) THEN	SWALLS	421
93	*****	SWALLS	422
94	* CHECK TO SEE IF THERE IS ENOUGH ARRAY SPACE	SWALLS	423
95	*****	SWALLS	424
96	IF(WTOT.GE.WMAX) THEN	SWALLS	425
97	WTOT = WMAX	SWALLS	426
98	PRINT *, 'DATA ENTRY ABORTED.'	SWALLS	427
99	PRINT *, 'MAXIMUM NUMBER OF DATA LINES IN FILE WOULD'	SWALLS	428
100	PRINT *, ' HAVE BEEN EXCEEDED. NO MORE THAN ',WMAX	SWALLS	429
101	PRINT *, ' DATA LINES ARE ALLOWED.'	SWALLS	430
102	PRINT *, ' TO INCREASE THE MAXIMUM NUMBER OF ENTRIES ALLOWED,'	SWALLS	431
103	PRINT *, ' CHANGE THE PARAMETER "WMAX" IN EACH COMMON OF'	SWALLS	432
104	PRINT *, ' EVERY SUBROUTINE (THERE ARE FOUR PLACES).'	SWALLS	433
105	PRINT *, ' THEN RECOMPILE THE PROGRAM.'	SWALLS	434
106	GOTO 10	SWALLS	435
107	ENDIF	SWALLS	436
108	*****	SWALLS	437
109	*	SWALLS	438
110	*** INDICATE EMPTY DATA FILE	SWALLS	439
111	IF (WTOT .EQ. 0) THEN	SWALLS	440
112	PRINT*	SWALLS	441
113	PRINT*, 'DATA FILE IS EMPTY !!!'	SWALLS	442
114	*	SWALLS	443
115	*** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE	SWALLS	444
116	ELSE	SWALLS	445
117	200 PRINT*	SWALLS	446
118	PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',	SWALLS	447
119	+ 'TO BE INSERTED'	SWALLS	448
120	PRINT*, ' (ENTER "0" TO ESCAPE "INSERTION" MODE) '	SWALLS	449
121	REWIND 1	SWALLS	450
122	READ(1,*,END=200) N	SWALLS	451
123	*	SWALLS	452
124	*** CHECK FOR VALID LINE NUMBER	SWALLS	453
125	IF ((N .LT. 0) .OR. (N .GT. WTOT)) THEN	SWALLS	454
126	PRINT*	SWALLS	455
127	PRINT*, 'INCORRECT NUMBER !!!'	SWALLS	456
128	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE',	SWALLS	457

129	+	'"INSERTION" MODE'	SWALLS	458
130		GOTO 200	SWALLS	459
131	*		SWALLS	460
132	***	ABORT INSERTION MODE	SWALLS	461
133		ELSE IF (N .EQ. 0) THEN	SWALLS	462
134		PRINT*	SWALLS	463
135		PRINT*, ' "INSERTION" MODE ABORTED'	SWALLS	464
136	*		SWALLS	465
137	***	MAKE ROOM FOR NEW LINE OF DATA	SWALLS	466
138		ELSE IF ((N .GT. 0) .AND. (N .LE. WTOT)) THEN	SWALLS	467
139		DO 230 X = WTOT,N,-1	SWALLS	468
140		DO 210 Y = 1,4	SWALLS	469
141		WALL(X+1,Y) = WALL(X,Y)	SWALLS	470
142	210	CONTINUE	SWALLS	471
143		DO 220 Y = 1,3	SWALLS	472
144		WDIM(X+1,Y) = WDIM(X,Y)	SWALLS	473
145	220	CONTINUE	SWALLS	474
146	230	CONTINUE	SWALLS	475
147	*		SWALLS	476
148	***	ENTER DATA FOR NEW LINE	SWALLS	477
149		WTOT = WTOT + 1	SWALLS	478
150		CALL DATAIN (1,N)	SWALLS	479
151	*		SWALLS	480
152	***	INITIALIZE FLAGS	SWALLS	481
153		OK1 = 0	SWALLS	482
154		OK2 = 0	SWALLS	483
155		OK = 0	SWALLS	484
156	*		SWALLS	485
157	***	TEST VALIDITY OF DATA	SWALLS	486
158	*		SWALLS	487
159	***	TEST IF NEW LAYER BELONGS TO THE NEXT WALL	SWALLS	488
160		IF ((WALL(N,1) .EQ. WALL(N+1,1))	SWALLS	489
161	+	.AND. (WALL(N,2) .EQ. WALL(N+1,2))	SWALLS	490
162	+	.AND. (WALL(N,3) .EQ. WALL(N+1,3))) THEN	SWALLS	491
163		IF ((WDIM(N,1) .EQ. WDIM(N+1,1))	SWALLS	492
164	+	.AND. (WDIM(N,2) .EQ. WDIM(N+1,2))) THEN	SWALLS	493
165		OK1 = 1	SWALLS	494
166		END IF	SWALLS	495
167		END IF	SWALLS	496
168	*		SWALLS	497
169	***	TEST IF NEW LAYER BELONGS TO PREVIOUS WALL	SWALLS	498
170		IF (N .GT. 1) THEN	SWALLS	499
171		IF ((WALL(N,1) .EQ. WALL(N-1,1))	SWALLS	500
172	+	.AND. (WALL(N,2) .EQ. WALL(N-1,2))	SWALLS	501
173	+	.AND. (WALL(N,3) .EQ. WALL(N-1,3))) THEN	SWALLS	502
174		IF ((WDIM(N,1) .EQ. WDIM(N-1,1))	SWALLS	503
175	+	.AND. (WDIM(N,2) .EQ. WDIM(N-1,2))) THEN	SWALLS	504
176		OK2 = 1	SWALLS	505
177		END IF	SWALLS	506
178		END IF	SWALLS	507
179		END IF	SWALLS	508
180	*		SWALLS	509
181		IF ((OK1 .EQ. 1) .OR. (OK2 .EQ. 1)) THEN	SWALLS	510
182		OK = 1	SWALLS	511
183		END IF	SWALLS	512
184	*		SWALLS	513
185		IF (OK .EQ. 1) THEN	SWALLS	514
186		PRINT*	SWALLS	515
187		PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N	SWALLS	516
188		CALL DISPLAY(N, COMMAND)	SWALLS	517
189	*		SWALLS	518
190	***	REJECT DATA IF DATA DOESN'T MATCH PREVIOUS OR NEXT LAYER	SWALLS	519
191		ELSE IF (OK .EQ. 0) THEN	SWALLS	520
192		PRINT*	SWALLS	521

193	PRINT*, 'YOUR DATA WAS NOT ACCEPTED !!!'	SWALLS	522
194	PRINT*, ' YOUR DATA MUST REPRESENT A LAYER ',	SWALLS	523
195	+ 'IN AN EXISTING WALL'	SWALLS	524
196	PRINT*, ' I.E. THE DIRECTION, FROM, TO, HEIGHT, AND ',	SWALLS	525
197	+ 'WIDTH'	SWALLS	526
198	PRINT*, ' PARAMETERS MUST MATCH THE WALL JUST ',	SWALLS	527
199	+ 'BEFORE'	SWALLS	528
200	PRINT*, ' OR JUST AFTER YOUR SPECIFIED INSERTION ',	SWALLS	529
201	+ 'POINT'	SWALLS	530
202	PRINT*	SWALLS	531
203	PRINT*, 'THE FOLLOWING DISPLAYS'	SWALLS	532
204	IF (N .GT. 1) PRINT*, 'THE LINE BEFORE YOUR LINE, '	SWALLS	533
205	PRINT*, 'YOUR LINE, AND THE LINE AFTER'	SWALLS	534
206	PRINT*	SWALLS	535
207	*	SWALLS	536
208	*** DISPLAY LINES OF DATA	SWALLS	537
209	IF (N .GT. 1) CALL DISPLAY (N-1, COMMAND)	SWALLS	538
210	CALL DISPLAY(N, COMMAND)	SWALLS	539
211	CALL DISPLAY (N+1, COMMAND)	SWALLS	540
212	*	SWALLS	541
213	*** REMOVE THE LINE OF INCORRECTLY ENTERED DATA	SWALLS	542
214	DO 270 X = N, WTOT	SWALLS	543
215	DO 250 Y = 1, 4	SWALLS	544
216	WALL(X,Y) = WALL(X+1,Y)	SWALLS	545
217	CONTINUE	SWALLS	546
218	DO 260 Y = 1, 3	SWALLS	547
219	WDIM(X,Y) = WDIM(X+1,Y)	SWALLS	548
220	CONTINUE	SWALLS	549
221	CONTINUE	SWALLS	550
222	WTOT = WTOT - 1	SWALLS	551
223	END IF	SWALLS	552
224	END IF	SWALLS	553
225	END IF	SWALLS	554
226	END IF	SWALLS	555
227	*	SWALLS	556
228	*-----	SWALLS	557
229	*** DELETE LINE ***	SWALLS	558
230	*-----	SWALLS	559
231	IF (COMMAND .EQ. 3) THEN	SWALLS	560
232	*	SWALLS	561
233	*** INDICATE EMPTY DATA FILE	SWALLS	562
234	IF (WTOT .EQ. 0) THEN	SWALLS	563
235	PRINT*	SWALLS	564
236	PRINT*, 'DATA FILE IS EMPTY !!!'	SWALLS	565
237	*	SWALLS	566
238	*** READ NUMBER OF LINE TO BE DELETED	SWALLS	567
239	ELSE	SWALLS	568
240	300 PRINT*	SWALLS	569
241	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'	SWALLS	570
242	PRINT*, ' (ENTER "0" TO ESCAPE DELETION MODE)'	SWALLS	571
243	REWIND 1	SWALLS	572
244	READ(1,*,END=300) N	SWALLS	573
245	*	SWALLS	574
246	*** CHECK VALIDITY OF LINE NUMBER	SWALLS	575
247	IF ((N .GT. WTOT) .OR. (N .LT. 0)) THEN	SWALLS	576
248	PRINT*	SWALLS	577
249	PRINT*, ' INCORRECT NUMBER !!!'	SWALLS	578
250	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE FROM',	SWALLS	579
251	+ '"DELETE" MODE'	SWALLS	580
252	GOTO 300	SWALLS	581
253	*	SWALLS	582
254	*** ABORT 'DELETE' MODE	SWALLS	583
255	ELSE IF (N .EQ. 0) THEN	SWALLS	584
256	PRINT*, ' "DELETE" MODE ABORTED'	SWALLS	585

257 *	SWALLS	586
258 *** DOUBLE CHECK CHOICE OF LINE TO BE DELETED	SWALLS	587
259 ELSE IF ((N .GT. 0) .AND. (N .LE. WTOT)) THEN	SWALLS	588
260 PRINT*	SWALLS	589
261 PRINT*, 'DOUBLE CHECK !!!'	SWALLS	590
262 PRINT*, ' DO YOU WANT TO DELETE THE FOLLOWING LINE?:'	SWALLS	591
263 CALL DISPLAY(N, COMMAND)	SWALLS	592
264 305 PRINT*, ' ENTER (1) YES OR (2) NO'	SWALLS	593
265 REWIND 1	SWALLS	594
266 READ(1,*,END=305) ANSWER	SWALLS	595
267 *	SWALLS	596
268 *** DELETE LINE	SWALLS	597
269 IF (ANSWER .EQ. 1) THEN	SWALLS	598
270 DO 330 X = N, WTOT - 1	SWALLS	599
271 DO 310 Y = 1,4	SWALLS	600
272 WALL(X,Y) = WALL(X+1,Y)	SWALLS	601
273 310 CONTINUE	SWALLS	602
274 DO 320 Y = 1,3	SWALLS	603
275 WDIM(X,Y) = WDIM(X+1,Y)	SWALLS	604
276 320 CONTINUE	SWALLS	605
277 330 CONTINUE	SWALLS	606
278 WTOT = WTOT - 1	SWALLS	607
279 PRINT*	SWALLS	608
280 PRINT*, 'LINE # ',N,' DELETED'	SWALLS	609
281 END IF	SWALLS	610
282 *	SWALLS	611
283 END IF	SWALLS	612
284 END IF	SWALLS	613
285 END IF	SWALLS	614
286 *	SWALLS	615
287 *-----	SWALLS	616
288 *** DISPLAY ALL DATA ***	SWALLS	617
289 *-----	SWALLS	618
290 IF (COMMAND .EQ. 4) THEN	SWALLS	619
291 *	SWALLS	620
292 *** INDICATE EMPTY DATA FILE	SWALLS	621
293 IF (WTOT .EQ. 0) THEN	SWALLS	622
294 PRINT*	SWALLS	623
295 PRINT*, 'DATA FILE IS EMPTY !!!'	SWALLS	624
296 *	SWALLS	625
297 *** DISPLAY DATA	SWALLS	626
298 ELSE	SWALLS	627
299 PRINT*	SWALLS	628
300 CALL DISPLAY(N, COMMAND)	SWALLS	629
301 *	SWALLS	630
302 END IF	SWALLS	631
303 END IF	SWALLS	632
304 *	SWALLS	633
305 *-----	SWALLS	634
306 *** ADD DATA ***	SWALLS	635
307 *-----	SWALLS	636
308 IF (COMMAND .EQ. 5) THEN	SWALLS	637
309 *	SWALLS	638
310 *** ENTER DATA	SWALLS	639
311 *****	SWALLS	640
312 * CHECK TO SEE IF THERE IS ENOUGH ARRAY SPACE	SWALLS	641
313 *****	SWALLS	642
314 500 IF(WTOT.GE.WMAX) THEN	SWALLS	643
315 WTOT = WMAX	SWALLS	644
316 PRINT *, 'DATA ENTRY ABORTED.'	SWALLS	645
317 PRINT *, 'MAXIMUM NUMBER OF DATA LINES IN FILE WOULD'	SWALLS	646
318 PRINT *, ' HAVE BEEN EXCEEDED. NO MORE THAN ',WMAX	SWALLS	647
319 PRINT *, ' DATA LINES ARE ALLOWED.'	SWALLS	648
320 PRINT *, ' TO INCREASE THE MAXIMUM NUMBER OF ENTRIES ALLOWED,'	SWALLS	649

321	PRINT *, ' CHANGE THE PARAMETER "WMAX" IN EACH COMMON OF '	SWALLS	650
322	PRINT *, ' EVERY SUBROUTINE (THERE ARE FOUR PLACES). '	SWALLS	651
323	PRINT *, ' THEN RECOMPILE THE PROGRAM. '	SWALLS	652
324	GOTO 10	SWALLS	653
325	ENDIF	SWALLS	654
326	*****	SWALLS	655
327	WTOT = WTOT + 1	SWALLS	656
328	CALL DATAIN (0,WTOT)	SWALLS	657
329	510 PRINT*	SWALLS	658
330	PRINT*, 'DO YOU WANT TO ENTER MORE DATA? (1) YES (2) NO'	SWALLS	659
331	PRINT*, ' ENTER A NUMBER !!!'	SWALLS	660
332	REWIND 1	SWALLS	661
333	READ(1,*,END=510) ANSWER	SWALLS	662
334	*	SWALLS	663
335	*** CHECK VALIDITY OF NUMBER	SWALLS	664
336	IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SWALLS	665
337	GOTO 510	SWALLS	666
338	*	SWALLS	667
339	*** ENTER MORE DATA	SWALLS	668
340	ELSE IF (ANSWER .EQ. 1) THEN	SWALLS	669
341	GOTO 500	SWALLS	670
342	*	SWALLS	671
343	*** DISCONTINUE DATA ENTRY	SWALLS	672
344	ELSE IF (ANSWER .EQ. 2) THEN	SWALLS	673
345	PRINT*	SWALLS	674
346	PRINT*, 'DATA ENTRY DISCONTINUED'	SWALLS	675
347	*	SWALLS	676
348	END IF	SWALLS	677
349	END IF	SWALLS	678
350	*	SWALLS	679
351	*-----	SWALLS	680
352	*** STORE DATA AND PROGRAM ***	SWALLS	681
353	*-----	SWALLS	682
354	IF (COMMAND .EQ. 6) THEN	SWALLS	683
355	600 PRINT*	SWALLS	684
356	PRINT*, 'DOUBLE CHECK !!!'	SWALLS	685
357	PRINT*, ' DO YOU YOU WANT TO STORE THIS DATA AND END PROG'	SWALLS	686
358	PRINT*, ' NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE '	SWALLS	687
359	PRINT*, ' OF THE SAME NAME !!!'	SWALLS	688
360	PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'	SWALLS	689
361	REWIND 1	SWALLS	690
362	READ(1,*,END=600) ANSWER	SWALLS	691
363	*	SWALLS	692
364	*** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM	SWALLS	693
365	IF (ANSWER .EQ. 1) THEN	SWALLS	694
366	QUIT = 1	SWALLS	695
367	RETURN	SWALLS	696
368	*	SWALLS	697
369	*** ABORT 'STORING' MODE	SWALLS	698
370	ELSE IF (ANSWER .EQ. 2) THEN	SWALLS	699
371	PRINT*	SWALLS	700
372	PRINT*, ' "STORING" MODE DISCONTINUED'	SWALLS	701
373	*	SWALLS	702
374	*** CHECK VALIDITY OF ANSWER	SWALLS	703
375	ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SWALLS	704
376	GOTO 600	SWALLS	705
377	*	SWALLS	706
378	ENDIF	SWALLS	707
379	ENDIF	SWALLS	708
380	*	SWALLS	709
381	*-----	SWALLS	710
382	*** END PROGRAM WITHOUT STORING DATA ***	SWALLS	711
383	*-----	SWALLS	712
384	IF (COMMAND .EQ. 7) THEN	SWALLS	713

385	700	PRINT*	SWALLS	714
386		PRINT*, 'DOUBLE CHECK !!!'	SWALLS	715
387		PRINT*, ' DO YOU WANT TO END THIS PROGRAM '	SWALLS	716
388	+	'WITHOUT STORING DATA?'	SWALLS	717
389		PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'	SWALLS	718
390		REWIND 1	SWALLS	719
391		READ(1,*,END=700) ANSWER	SWALLS	720
392	*		SWALLS	721
393	***	SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM	SWALLS	722
394		IF (ANSWER .EQ. 1) THEN	SWALLS	723
395		ABORT = 1	SWALLS	724
396		RETURN	SWALLS	725
397	*		SWALLS	726
398	***	ABORT 'STORING' MODE	SWALLS	727
399		ELSE IF (ANSWER .EQ. 2) THEN	SWALLS	728
400		PRINT*	SWALLS	729
401		PRINT*, ' "ABORTION" MODE DISCONTINUED'	SWALLS	730
402	*		SWALLS	731
403	***	CHECK VALIDITY OF ANSWER	SWALLS	732
404		ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SWALLS	733
405		GOTO 700	SWALLS	734
406	*		SWALLS	735
407		ENDIF	SWALLS	736
408		ENDIF	SWALLS	737
409	*		SWALLS	738
410	*	-----	SWALLS	739
411	***	LOOP TO BEGINNING OF 'MANIP' SUBROUTINE	SWALLS	740
412	*	-----	SWALLS	741
413		GOTO 10	SWALLS	742
414	*		SWALLS	743
415		END	SWALLS	744

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	2	DUMMY-ARG	INTEGER	
ANSWER	2227B		INTEGER	
COMMAND	2237B		INTEGER	
DIR	NONE	UNUSED/*S*	CHAR*3	
DOK	NONE	UNUSED/*S*	INTEGER	
FLAG1	2230B		INTEGER	
FROM	NONE	UNUSED/*S*	CHAR*3	
INSERT	NONE	UNUSED/*S*	INTEGER	
LOK	NONE	UNUSED/*S*	INTEGER	
MAT	NONE	UNUSED/*S*	CHAR*3	
N	2231B		INTEGER	
NOK	NONE	UNUSED/*S*	INTEGER	
OK	2232B		INTEGER	
OK1	2233B		INTEGER	
OK2	2234B		INTEGER	
QUIT	1	DUMMY-ARG	INTEGER	
TEMP	NONE	UNUSED/*S*	INTEGER	
TO	NONE	UNUSED/*S*	CHAR*3	
V	NONE	UNUSED/*S*	INTEGER	
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	
X	2235B		INTEGER	
Y	2236B		INTEGER	

FTN 5.1+552 83/12/24. 11.29.46 PAGE 18
 SUBROUTINE MANIP 74/175 OPT=0

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

WMAX INTEGER 75

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

DATAIN 2 SUBROUTINE
 DISPLAY 2 SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

10	7B	35	300	611B	240
100	50B	61	305	660B	264
200	161B	117	310	INACTIVE DO-TERM	273
210	INACTIVE DO-TERM	142	320	INACTIVE DO-TERM	276
220	INACTIVE DO-TERM	145	330	INACTIVE DO-TERM	277
230	INACTIVE DO-TERM	146	500	1003B	314
250	INACTIVE DO-TERM	217	510	1035B	329
260	INACTIVE DO-TERM	220	600	1076B	355
270	INACTIVE DO-TERM	221	700	1150B	385

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

MANIP 5B 2

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE1 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	2253B = 1195
CM LABELLED COMMON LENGTH	475B = 317
CM STORAGE USED	65000B = 27136
COMPILE TIME	0.625 SECONDS

1	SUBROUTINE DISPLAY (LINE, COMMAND)	SWALLS	745
2	*****	COMW	1
3	*** COMMON FOR DATABASE OF WALL PARAMETERS	***COMW	2
4	*****	COMW	3
5	INTEGER WMAX	COMW	4
6	PARAMETER (WMAX = 75)	COMW	5
7	COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR	COMW	6
8	COMMON /WALLC/ WALL(WMAX,4)	COMW	7
9	INTEGER WTOT,WERR	COMW	8
10	REAL WDIM	COMW	9
11	CHARACTER *3 WALL	COMW	10
12	* =====	COMW	11
13	** DESCRIPTION OF ARRAYS	COMW	12
14	* =====	COMW	13
15	* WALL IDENTIFICATION	COMW	14
16	* -----	COMW	15
17	* DIRECTION FROM TO	COMW	16
18	* ROOM ROOM	COMW	17
19	* -----	COMW	18
20	* WALL(X,1) WALL(X,2) WALL(X,3)	COMW	19
21	* A3 A3 A3	COMW	20
22	* =====	COMW	21
23	* WALL PARAMETERS	COMW	22
24	* -----	COMW	23
25	* MATERIAL HEIGHT WIDTH LAYER THICKNESS	COMW	24
26	* -----	COMW	25
27	* WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3)	COMW	26
28	* A3 F8.2 F8.2 F8.2	COMW	27
29	*****	COMW	28
30	*****	COMW	29
31	INTEGER LINE, COMMAND, N	SWALLS	747
32	1000 FORMAT (8(3X,A))	SWALLS	748
33	2000 FORMAT (4X,13,8X,A3,7X,A3,3X,A3,2X,F6.2,2X,F6.2,5X,F6.2,7X,A3)	SWALLS	749
34	PRINT 1000, 'LINE #', 'DIRECTION', 'FROM', 'TO', 'HEIGHT',	SWALLS	750
35	+ 'WIDTH', 'THICKNESS', 'MATERIAL'	SWALLS	751
36	IF (COMMAND .EQ 4) THEN	SWALLS	752
37	DO 10 N = 1,WTOT	SWALLS	753
38	PRINT 2000, N,WALL(N,1),WALL(N,2),WALL(N,3),WDIM(N,1),	SWALLS	754
39	+ WDIM(N,2),WDIM(N,3),WALL(N,4)	SWALLS	755
40	10 CONTINUE	SWALLS	756
41	ELSE	SWALLS	757
42	PRINT 2000, LINE,WALL(LINE,1),WALL(LINE,2),WALL(LINE,3),	SWALLS	758
43	+ WDIM(LINE,1),WDIM(LINE,2),WDIM(LINE,3), WALL(LINE,4)	SWALLS	759
44	END IF	SWALLS	760
45	RETURN	SWALLS	761
46	END	SWALLS	762

--VARIABLE MAP--(LO=A)
--NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

COMMAND	2	DUMMY-ARG	INTEGER	
LINE	1	DUMMY-ARG	INTEGER	
N	244B		INTEGER	
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

WMAX INTEGER 75

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

10	INACTIVE	DO-TERM	40
1000	150B	FORMAT	32
2000	152B	FORMAT	33

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

DISPLAY 5B 2

--STATISTICS--

PROGRAM-UNIT LENGTH	250B = 168
CM LABELLED COMMON LENGTH	475B = 317
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.085 SECONDS

1	INTEGER FUNCTION VAL(String)	SWALLS	763
2	C** RETURNS THE INTEGER VALUE OF A STRING.	SWALLS	764
3	INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN	SWALLS	765
4	CHARACTER * (*) STRING	SWALLS	766
5	L = GETLEN(String)	SWALLS	767
6	NUMBER = 0	SWALLS	768
7	DO 10 X = L,1,-1	SWALLS	769
8	EXP = L - X	SWALLS	770
9	DIGIT = ICHAR(String(X:X)) - 16	SWALLS	771
10	NUMBER = NUMBER + DIGIT*10**EXP	SWALLS	772
11 10	CONTINUE	SWALLS	773
12	VAL = NUMBER	SWALLS	774
13	RETURN	SWALLS	775
14	END	SWALLS	776

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

DIGIT	76B		INTEGER
EXP	75B		INTEGER
L	74B		INTEGER
NUMBER	72B		INTEGER
STRING	1	DUMMY-ARG	CHAR*(*)
VAL	71B		INTEGER
X	73B		INTEGER

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	11
----	----------	---------	----

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS---ARGS---

VAL	6B	1
-----	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.039 SECONDS

1	INTEGER FUNCTION GETLEN (STRING)	SWALLS	777
2	C	SWALLS	778
3	C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING	SWALLS	779
4	C	SWALLS	780
5	C	SWALLS	781
6	C ARGUMENT DEFINITIONS --	SWALLS	782
7	C READ ARGUMENTS	SWALLS	783
8	C STRING - STRING WHOSE LENGTH IS TO BE DETERMINED	SWALLS	784
9	C	SWALLS	785
10	CHARACTER * (*) STRING	SWALLS	786
11	C	SWALLS	787
12	C FUNCTION PARAMETERS	SWALLS	788
13	CHARACTER * 1 BLANK	SWALLS	789
14	PARAMETER (BLANK = ' ')	SWALLS	790
15	C	SWALLS	791
16	C LOCAL VARIABLES	SWALLS	792
17	INTEGER NEXT	SWALLS	793
18	C	SWALLS	794
19	C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK	SWALLS	795
20	DO 10 NEXT = LEN(STRING),1,-1	SWALLS	796
21	IF (STRING(NEXT : NEXT) .NE. BLANK) THEN	SWALLS	797
22	GETLEN = NEXT	SWALLS	798
23	RETURN	SWALLS	799
24	ENDIF	SWALLS	800
25	10 CONTINUE	SWALLS	801
26	C	SWALLS	802
27	C ALL CHARACTERS ARE BLANKS	SWALLS	803
28	GETLEN = 0	SWALLS	804
29	C	SWALLS	805
30	RETURN	SWALLS	806
31	END	SWALLS	807

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

GETLEN	63B		INTEGER
NEXT	64B		INTEGER
STRING	1	DUMMY-ARG	CHAR*(*)

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME---TYPE-----VALUE

BLANK	CHAR*1		
-------	--------	--	--

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

LEN	INTEGER	1	INTRINSIC
-----	---------	---	-----------

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	25
----	----------	---------	----

--ENTRY POINTS--(LO=A)
-NAME--ADDRESS--ARGS--

GETLEN 6B 1

--STATISTICS--

PROGRAM-UNIT LENGTH	70B = 56
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.037 SECONDS

```

1      SUBROUTINE LWALL                                LWALL      1
2      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LWALL      2
3      *!!!!                                           !!!LWALL      3
4      *!!!!   LOAD THE CONTENTS OF THE FILE 'WALLS' INTO ARRAYS WALL AND WDIM. LWALL      4
5      *!!!!                                           !!!LWALL      5
6      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LWALL      6
7      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LWALL      7
8      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!COMW      1
9      ***   COMMON FOR DATABASE OF WALL PARAMETERS                                ***COMW      2
10     *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!COMW      3
11     INTEGER WMAX                                     COMW      4
12     PARAMETER (WMAX = 75)                           COMW      5
13     COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR          COMW      6
14     COMMON /WALLC/ WALL(WMAX,4)                     COMW      7
15     INTEGER WTOT,WERR                                COMW      8
16     REAL WDIM                                         COMW      9
17     CHARACTER *3 WALL                                COMW     10
18     * ===== COMW     11
19     ** DESCRIPTION OF ARRAYS COMW     12
20     * ===== COMW     13
21     *   WALL IDENTIFICATION COMW     14
22     * ----- COMW     15
23     * DIRECTION      FROM      TO COMW     16
24     *                ROOM      ROOM COMW     17
25     * ----- COMW     18
26     * WALL(X,1)  WALL(X,2)  WALL(X,3) COMW     19
27     *   A3      A3      A3 COMW     20
28     * ===== COMW     21
29     *   WALL PARAMETERS COMW     22
30     * ----- COMW     23
31     * MATERIAL      HEIGHT      WIDTH      LAYER THICKNESS COMW     24
32     * ----- COMW     25
33     * WALL(X,4)  WDIM(X,1)  WDIM(X,2)  WDIM(X,3) COMW     26
34     *   A3      F8.2      F8.2      F8.2 COMW     27
35     * ===== COMW     28
36     * ===== COMW     29
37     * ===== COMF      1
38     *** COMMON FOR INITIAL PARAMETERS                                ***COMF      2
39     * ===== COMF      3
40     INTEGER FMAX                                     COMF      4
41     PARAMETER (FMAX = 50)                           COMF      5
42     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
43     $          FTOT COMF      7
44     COMMON /INITILC/ BLDG COMF      8
45     CHARACTER * 5 BLDG COMF      9
46     REAL FREQ, AFLAG, RFLAG, FREQA COMF     10
47     INTEGER QUALITY, FERR, FTOT COMF     11
48     * ===== COMF     12
49     * ===== COMF     13
50     * ===== LWALL     10
51     *   DECLARATION OF VARIABLES LWALL     11
52     * ===== LWALL     12
53     INTEGER GETLEN, R, C LWALL     13
54     CHARACTER * 7 NAME, PFN LWALL     14
55     * ===== LWALL     15
56     * LWALL     16
57     * ===== LWALL     17
58     NAME = 'B'//BLDG(1:GETLEN(BLDG))// 'W' LWALL     18
59     PFN = NAME (1:GETLEN(NAME)) LWALL     19
60     WERR = 0 LWALL     20
61     CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',WERR) LWALL     21
62     IF ( WERR .EQ. 0 ) THEN LWALL     22
63     OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED', LWALL     23
64     $      STATUS='OLD', ACCESS='SEQUENTIAL') LWALL     24

```


65	1000	FORMAT (1X,4(1X,A3),3(1X,F8.2))	LWALL	25
66		WTOT = 0	LWALL	26
67		DO 10 R = 1,WMAX	LWALL	27
68		READ (3,1000,END=20)(WALL(R,C),C=1,4),(WDIM(R,C),C=1,3)	LWALL	28
69		WTOT = WTOT + 1	LWALL	29
70	10	CONTINUE	LWALL	30
71	20	CONTINUE	LWALL	31
72		CLOSE(3,STATUS='DELETE')	LWALL	32
73		ELSE IF (WERR .EQ. 2) THEN	LWALL	33
74		CALL WARNING (7)	LWALL	34
75		ELSE	LWALL	35
76		CALL WARNING (8)	LWALL	36
77		END IF	LWALL	37
78		RETURN	LWALL	38
79		END	LWALL	39

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	255B		INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
NAME	256B		CHAR*7	
PFN	257B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
R	254B		INTEGER	
RFLAG	3B	/INITILN/	REAL	
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES---DEF

10	INACTIVE	DO-TERM	70
20	117B		71
1000	155B	FORMAT	65

FTN 5.1+552 83/12/24. 11.29.46 PAGE 26
SUBROUTINE LWALL 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS---ARGS---

LWALL 5B 0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE3 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	265B = 181
CM LABELLED COMMON LENGTH	566B = 374
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.105 SECONDS

1	SUBROUTINE WARNING(ERR)	WARNING	1
2	INTEGER ERR, ERRM	WARNING	2
3	CHARACTER*45 MESSAGE(20)	WARNING	3
4	DATA MESSAGE(1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	4
5	DATA MESSAGE(2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE	WARNING	5
6	DATA MESSAGE(3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG	WARNING	6
7	DATA MESSAGE(4)/'FILE HANDLING PROBLEM ON "MATTER" FILE	WARNING	7
8	DATA MESSAGE(5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	8
9	DATA MESSAGE(6)/'FILE HANDLING PROBLEM ON "TYPE" FILE	WARNING	9
10	DATA MESSAGE(7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	10
11	DATA MESSAGE(8)/'FILE HANDLING PROBLEM ON "WALL" FILE	WARNING	11
12	DATA MESSAGE(9)/'HEIGHT AND WIDTH OF ROOM MISSING	WARNING	12
13	DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING	WARNING	13
14	DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG	WARNING	14
15	DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE	WARNING	15
16	DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE	WARNING	16
17	DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE	WARNING	17
18	DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE	WARNING	18
19	DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE	WARNING	19
20	DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE	WARNING	20
21	DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE	WARNING	21
22	DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE	WARNING	22
23	DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE	WARNING	23
24	ERRM=12	WARNING	24
25	IERR = ERR	WARNING	25
26	IF(ERR.GT.ERRM) IERR=20	WARNING	26
27	WRITE(6,20)	WARNING	27
28	WRITE(6,10) ERR,MESSAGE(IERR)	WARNING	28
29	WRITE(6,20)	WARNING	29
30 10	FORMAT(' ***WARNING NUMBER = ',I5,' *** ',A45)	WARNING	30
31 20	FORMAT(' ')	WARNING	31
32	RETURN	WARNING	32
33	END	WARNING	33

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ERR	1	DUMMY-ARG	INTEGER	
ERRM	60B		INTEGER	
IERR	213B		INTEGER	
MESSAGE	61B		CHAR*45	20

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	34B	FORMAT	30
20	42B	FORMAT	31

FTN 5.1+552 83/12/24. 11.29.46 PAGE 28
SUBROUTINE WARNING 74/175 OPT=0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARCS---

WARNING	5B	1
---------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	216B = 142
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.060 SECONDS

1	SUBROUTINE ERROR(IERR)	ERROR	1
2	CHARACTER*45 MESSAGE(20)	ERROR	2
3	DATA MESSAGE(1) //'MATERIALS DATA BASE IS EMPTY	// ERROR	3
4	DATA MESSAGE(2) //'FREQUENCY IS OUT OF RANGE	// ERROR	4
5	DATA MESSAGE(3) //'THIS MATERIAL IS NOT IN DATA BASE	// ERROR	5
6	DATA MESSAGE(4) //'DENOMINATOR IS ZERO	// ERROR	6
7	DATA MESSAGE(5) //'FILE HANDLING ERROR	// ERROR	7
8	DATA MESSAGE(6) //'ERROR CODE IS OUT OF RANGE	// ERROR	8
9	DATA MESSAGE(7) //'ERROR CODE IS OUT OF RANGE	// ERROR	9
10	DATA MESSAGE(8) //'ERROR CODE IS OUT OF RANGE	// ERROR	10
11	DATA MESSAGE(9) //'ERROR CODE IS OUT OF RANGE	// ERROR	11
12	DATA MESSAGE(10) //'ERROR CODE IS OUT OF RANGE	// ERROR	12
13	DATA MESSAGE(11) //'ERROR CODE IS OUT OF RANGE	// ERROR	13
14	DATA MESSAGE(12) //'ERROR CODE IS OUT OF RANGE	// ERROR	14
15	DATA MESSAGE(13) //'ERROR CODE IS OUT OF RANGE	// ERROR	15
16	DATA MESSAGE(14) //'ERROR CODE IS OUT OF RANGE	// ERROR	16
17	DATA MESSAGE(15) //'ERROR CODE IS OUT OF RANGE	// ERROR	17
18	DATA MESSAGE(16) //'ERROR CODE IS OUT OF RANGE	// ERROR	18
19	DATA MESSAGE(17) //'ERROR CODE IS OUT OF RANGE	// ERROR	19
20	DATA MESSAGE(18) //'ERROR CODE IS OUT OF RANGE	// ERROR	20
21	DATA MESSAGE(19) //'ERROR CODE IS OUT OF RANGE	// ERROR	21
22	DATA MESSAGE(20) //'ERROR CODE IS OUT OF RANGE	// ERROR	22
23	IERRM=5	ERROR	23
24	IF(IERR.GT.IERRM) IERR=20	ERROR	24
25	WRITE(6,10) IERR,MESSAGE(IERR)	ERROR	25
26	10 FORMAT(' ***ERROR NUMBER = ',I5,' *** ',A45)	ERROR	26
27	CALL PMDSTOP	ERROR	27
28	STOP 'ERROR'	ERROR	28
29	END	ERROR	29

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

IERR	1	DUMMY-ARG	INTEGER	
IERRM	210B		INTEGER	
MESSAGE	56B		CHAR*45	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PMDSTOP		0	SUBROUTINE
---------	--	---	------------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

10	36B	FORMAT	26
----	-----	--------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

ERROR	5B	1
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	213B = 139
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.056 SECONDS

Appendix 9.5 Listing of Computer Program STYPES

1	PROGRAM STYPES (INPUT,TAPE1=INPUT)	STYPES	1
2	*	STYPES	2
3	*THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH TYPE	STYPES	3
4	*IN THE BUILDING AND STORES IT. THE FILE NAME IS CREATED BY	STYPES	4
5	*ATTACHING "B" TO THE FRONT OF AND "W" TO THE BACK OF THE BUILDING	STYPES	5
6	*IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE	STYPES	6
7	*THAN 5 ALPHANUMERIC CHARACTERS.	STYPES	7
8		STYPES	8
9	*****COMF		1
10	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
11	*****COMF		3
12	INTEGER FMAX	COMF	4
13	PARAMETER (FMAX = 50)	COMF	5
14	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
15	5 FTOT	COMF	7
16	COMMON /INITILC/ BLDG	COMF	8
17	CHARACTER * 5 BLDG	COMF	9
18	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
19	INTEGER QUALITY, FERR, FTOT	COMF	11
20	*****COMF		12
21	*****COMF		13
22	*****COMR		1
23	*** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS	***COMR	2
24	*****COMR		3
25	INTEGER RMAX	COMR	4
26	PARAMETER (RMAX = 20)	COMR	5
27	COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)	COMR	6
28	INTEGER NROOMS	COMR	7
29	REAL ROOM	COMR	8
30	*****COMR		9
31	*****COMR		10
32	*****COMT		1
33	*** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS	***COMT	2
34	*****COMT		3
35	INTEGER TMAX	COMT	4
36	PARAMETER (TMAX=35)	COMT	5
37	COMMON /TYPEN/ TDIM(TMAX, 4), TTOT, TDB2(TMAX, 2), TDBTOT, TERR	COMT	6
38	COMMON /TYPEC/ TYPE(TMAX, 3), TDB1(TMAX)	COMT	7
39	INTEGER TTOT, TDBTOT, TERR	COMT	8
40	REAL TDIM, TDB2	COMT	9
41	CHARACTER * 3 TYPE, TDB1	COMT	10
42	*=====	COMT	11
43	* DESCRIPTION OF ARRAYS	COMT	12
44	*=====	COMT	13
45	* ID MATERIAL FRAME MATERIAL	COMT	14
46	*-----	COMT	15
47	*TYPE(X,1) TYPE(X,2) TYPE(X,3)	COMT	16
48	* A3 A3 A3	COMT	17
49	*=====	COMT	18
50	* HEIGHT WIDTH LAYER DISTANCE	COMT	19
51	* THICKNESS ABOVE FLOOR	COMT	20
52	*-----	COMT	21
53	* TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4)	COMT	22
54	* F8.2 F8.2 F8.2 F8.2	COMT	23
55	*=====	COMT	24
56	* ID ATTENUATION AREA	COMT	25
57	*-----	COMT	26
58	* TDB1(X) TDB2(X,1) TDB2(X,2)	COMT	27
59	* A3 E9.3 E9.3	COMT	28
60	*****COMT		29
61	*****COMT		30
62	INTEGER GETLEN, QUIT, ABORT, ANSWER, OLDFILE, N, Y1, Y2, LINE	STYPES	12
63	INTEGER IERR	STYPES	13
64	CHARACTER * 7 PFN	STYPES	14

65 *	STYPES	15
66 * INITIALIZATION	STYPES	16
67 QUIT = 0	STYPES	17
68 TTOT = 0	STYPES	18
69 ABORT = 0	STYPES	19
70 100 PRINT*	STYPES	20
71 PRINT *, 'ENTER BUILDING IDENTIFICATION (E.G. '101')'	STYPES	21
72 PRINT *, ' (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'	STYPES	22
73 REWIND 1	STYPES	23
74 READ(1,*,END=100) BLDG	STYPES	24
75	STYPES	25
76 IF (GETLEN(BLDG) .GT. 5) THEN	STYPES	26
77 GO TO 100	STYPES	27
78 END IF	STYPES	28
79 PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'T'	STYPES	29
80 *	STYPES	30
81 *** LOAD DATA ID EXISTING FILE IF NECESSARY	STYPES	31
82 200 PRINT*	STYPES	32
83 PRINT*, 'WILL THIS BE'	STYPES	33
84 PRINT*, ' (1) A MODIFICATION OF AN EXISTING FILE?'	STYPES	34
85 PRINT*, ' (2) A NEW FILE?'	STYPES	35
86 PRINT*, 'ENTER A NUMBER !!!'	STYPES	36
87 REWIND 1	STYPES	37
88 READ(1,*,END=200) OLDFILE	STYPES	38
89 IF ((OLDFILE .NE. 1) .AND. (OLDFILE .NE. 2)) THEN	STYPES	39
90 GOTO 200	STYPES	40
91 ELSE IF (OLDFILE .EQ. 1) THEN	STYPES	41
92 *	STYPES	42
93 *** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	STYPES	43
94 IERR = 0	STYPES	44
95 CALL PF ('GET',0,PFN(1:GETLEN(PFN)), 'RC', IERR)	STYPES	45
96 IF (IERR .EQ. 2) THEN	STYPES	46
97 PRINT*	STYPES	47
98 PRINT *, 'FILE ', PFN, ' NOT FOUND'	STYPES	48
99 PRINT*, 'PROGRAM ABORTED!!!'	STYPES	49
100 PRINT*	STYPES	50
101 PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART ',	STYPES	51
102 + 'PROGRAM'	STYPES	52
103 PRINT*	STYPES	53
104 STOP	STYPES	54
105 *	STYPES	55
106 ELSE	STYPES	56
107 CALL LTYPE	STYPES	57
108 IF (TERR .NE. 0) CALL ERROR(5)	STYPES	58
109 END IF	STYPES	59
110 ELSE IF (OLDFILE .EQ. 2) THEN	STYPES	60
111 *	STYPES	61
112 *** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	STYPES	62
113 IERR = 0	STYPES	63
114 CALL PF ('GET',0,PFN(1:GETLEN(PFN)), 'RC', IERR)	STYPES	64
115 IF (IERR .EQ. 0) THEN	STYPES	65
116 PRINT*	STYPES	66
117 PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ', BLDG	STYPES	67
118 PRINT*	STYPES	68
119 PRINT*, 'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',	STYPES	69
120 + 'OVER THE OLD FILE.'	STYPES	70
121 250 PRINT*	STYPES	71
122 PRINT*, 'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'	STYPES	72
123 PRINT*, 'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'	STYPES	73
124 REWIND 1	STYPES	74
125 READ(1,*,END=250) ANSWER	STYPES	75
126 PRINT*	STYPES	76
127 PRINT*, 'PROGRAM HAS BEEN ABORTED, PER YOUR REQUEST'	STYPES	77
128 PRINT*	STYPES	78

129	IF (ANSWER .EQ. 1) THEN	STYPES	77
130	STOP	STYPES	80
131	ELSE IF (ANSWER .EQ. 2) THEN	STYPES	81
132 255	CONTINUE	STYPES	82
133	ELSE	STYPES	83
134	GOTO 250	STYPES	84
135	END IF	STYPES	85
136	ELSE IF (IERR .EQ. 2) THEN	STYPES	86
137 *		STYPES	87
138 *	NO DATA FILE EXISTS FOR THIS BUILDING AND DATA ENTRY	STYPES	88
139 *	CAN CONTINUE	STYPES	89
140 *		STYPES	90
141 260	CONTINUE	STYPES	91
142	ELSE	STYPES	92
143 *		STYPES	93
144 *	**PERMANENT FILE ERROR	STYPES	94
145 *		STYPES	95
146	PRINT*	STYPES	96
147	PRINT*, 'PROGRAM ABORTED !!!'	STYPES	97
148	PRINT*, ' SOME PERMANENT FILE ERROR HAS OCCURRED'	STYPES	98
149	PRINT*, ' DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',	STYPES	99
150 +	'AND TRY AGAIN'	STYPES	100
151	STOP	STYPES	101
152	END IF	STYPES	102
153	PRINT*	STYPES	103
154	PRINT*, ' BEGIN ENTERING DATA'	STYPES	104
155 300	TTOT = TTOT + 1	STYPES	105
156	IF (TTOT .EQ. 1) THEN	STYPES	106
157	CALL DATAIN(1,TTOT)	STYPES	107
158	ELSE	STYPES	108
159	CALL DATAIN (0,TTOT)	STYPES	109
160	END IF	STYPES	110
161 400	PRINT*	STYPES	111
162	PRINT*, 'DO YOU WANT TO ENTER MORE DATA?',	STYPES	112
163 +	'(1) YES (2) NO'	STYPES	113
164	PRINT*, ' ENTER A NUMBEER !!!'	STYPES	114
165	REWIND 1	STYPES	115
166	READ(1,*,END=400) ANSWER	STYPES	116
167	IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	STYPES	117
168	GOTO 400	STYPES	118
169	ELSE IF (ANSWER .EQ. 1) THEN	STYPES	119
170	GOTO 300	STYPES	120
171	ELSE IF (ANSWER .EQ. 2) THEN	STYPES	121
172	PRINT*	STYPES	122
173	PRINT*, 'DATA ENTRY DISCONTINUED'	STYPES	123
174	END IF	STYPES	124
175	END IF	STYPES	125
176 *		STYPES	126
177 ***	MANIPULATE DATA	STYPES	127
178	CALL MANIP (QUIT,ABORT)	STYPES	128
179 *		STYPES	129
180 ***	TERMINATE PROGRAM, STORING DATA IF NECESSARY	STYPES	130
181	IF (QUIT .EQ. 1) THEN	STYPES	131
182	OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',	STYPES	132
183 +	ACCESS='SEQUENTIAL',STATUS='NEW')	STYPES	133
184 500	FORMAT (1X,3(1X,A3),4(1X,F8.2))	STYPES	134
185	DO 600 N = 1,TTOT	STYPES	135
186	WRITE (6,500)(TYPE(N,Y1), Y1=1,3),(TDIM(N,Y2), Y2=1,4)	STYPES	136
187 600	CONTINUE	STYPES	137
188	ENDFILE(6)	STYPES	138
189	CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))	STYPES	139
WARNING*	NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT		
190	CLOSE(6,STATUS='DELETE')	STYPES	140
191	PRINT*	STYPES	141

192	PRINT*, 'DATA HAS BEEN STORED AND PROGRAM TERMINATED'	STYPES	142
193	END IF	STYPES	143
194	IF(ABORT .EQ. 1) THEN	STYPES	144
195	PRINT*	STYPES	145
196	PRINT*, 'PROGRAM HAS BEEN ABORTED'	STYPES	146
197	PRINT*, ' NO DATA HAS BEEN STORED !!!'	STYPES	147
198	END IF	STYPES	148
199	STOP	STYPES	149
200	END	STYPES	150

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	1077B		INTEGER	
AFLAG	2B	/INITILN/	REAL	
ANSWER	1100B		INTEGER	
BLDG	0B	/INITILC/	CHAR*5	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
IERR	1105B		INTEGER	
LINE	NONE	UNUSED/*S*	INTEGER	
N	1102B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
OLDFILE	1101B		INTEGER	
PFN	1106B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
QUIT	1076B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPEPC/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPEPC/	CHAR*3	105
Y1	1103B		INTEGER	
Y2	1104B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
RMAX	INTEGER	20
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS----- -NAME-----TYPE-----ARGS-----CLASS-----

DATAIN		2	SUBROUTINE	LTYPE	0	SUBROUTINE
ERROR		1	SUBROUTINE	MANIP	2	SUBROUTINE
GETLEN	INTEGER	1	FUNCTION	PF	5	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF			-LABEL-ADDRESS-----PROPERTIES-----DEF		
100	21B	70	300	246B	155
200	47B	82	400	260B	161
250	166B	121	500	626B	184
255	*NO REFS*	132	600	INACTIVE	187
260	*NO REFS*	141			

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

STYPES 14B 0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE1 FMT/SEQ
TAPE6 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH 1112B = 586
CM LABELLED COMMON LENGTH 1761B = 1009
CM STORAGE USED 63000B = 26112
COMPILE TIME 0.287 SECONDS

1 WARNING ERROR IN STYPES

1	SUBROUTINE DATAIN (INSERT,LINE)	STYPES	151
2	*****COMR		1
3	*** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR		2
4	*****COMR		3
5	INTEGER RMAX	COMR	4
6	PARAMETER (RMAX = 20)	COMR	5
7	COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)	COMR	6
8	INTEGER NROOMS	COMR	7
9	REAL ROOM	COMR	8
10	*****COMR		9
11	*****COMR		10
12	*****COMT		1
13	*** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT		2
14	*****COMT		3
15	INTEGER TMAX	COMT	4
16	PARAMETER (TMAX=35)	COMT	5
17	COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR	COMT	6
18	COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)	COMT	7
19	INTEGER TTOT,TDBTOT,TERR	COMT	8
20	REAL TDIM,TDB2	COMT	9
21	CHARACTER * 3 TYPE,TDB1	COMT	10
22	*=====	COMT	11
23	* DESCRIPTION OF ARRAYS	COMT	12
24	*=====	COMT	13
25	* ID MATERIAL FRAME MATERIAL	COMT	14
26	*-----	COMT	15
27	*TYPE(X,1) TYPE(X,2) TYPE(X,3)	COMT	16
28	* A3 A3 A3	COMT	17
29	*=====	COMT	18
30	* HEIGHT WIDTH LAYER DISTANCE	COMT	19
31	* THICKNESS ABOVE FLOOR	COMT	20
32	*-----	COMT	21
33	* TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4)	COMT	22
34	* F8.2 F8.2 F8.2 F8.2	COMT	23
35	*=====	COMT	24
36	* ID ATTENUATION AREA	COMT	25
37	*-----	COMT	26
38	* TDB1(X) TDB2(X,1) TDB2(X,2)	COMT	27
39	* A3 E9.3 E9.3	COMT	28
40	*****COMT		29
41	*****COMT		30
42	INTEGER ANSWER,LOK,DOK,NOK,GETLEN,VAL, INSERT,LINE,V	STYPES	154
43	REAL H,W,T,ABOVE	STYPES	155
44	CHARACTER *3 DIR,ID,MAT,FMAT	STYPES	156
45	99 IF (INSERT .EQ. 1) THEN	STYPES	157
46	ANSWER = 1	STYPES	158
47	INSERT = 1	STYPES	159
48	ELSE	STYPES	160
49	100 PRINT*	STYPES	161
50	PRINT*, 'IS THIS THE FIRST LAYER OF A DOOR OR WINDOW?'	STYPES	162
51	+ ' (1) YES (2) NO'	STYPES	163
52	PRINT*, ' ENTER "0" TO ESCAPE "DATA ENTRY" MODE'	STYPES	164
53	PRINT*, ' ENTER A NUMBER!!!'	STYPES	165
54	REWIND 1	STYPES	166
55	READ(1,*,END=100) ANSWER	STYPES	167
56	END IF	STYPES	168
57	*	STYPES	169
58	IF (ANSWER .EQ. 0) THEN	STYPES	170
59	TTOT = TTOT - 1	STYPES	171
60	PRINT*	STYPES	172
61	PRINT*, 'DATA ENTRY MODE ABORTED'	STYPES	173
62	END IF	STYPES	174
63	*	STYPES	175
64	IF ((ANSWER .NE. 2)	STYPES	176

65	+	.AND. (ANSWER .NE. 1)	STYPES	177
66	+	.AND. (ANSWER .NE. 0)) THEN	STYPES	178
67		PRINT*	STYPES	179
68		PRINT*, 'INCORRECT NUMBER!!'	STYPES	180
69		PRINT*, ' TRY AGAIN!! -OR- ENTER "0" TO ESCAPE DATA ENTRY MODE'	STYPES	181
70		GOTO 99	STYPES	182
71		END IF	STYPES	183
72	*		STYPES	184
73	*		STYPES	185
74		IF (ANSWER .EQ. 1) THEN	STYPES	186
75	300	PRINT*	STYPES	187
76		PRINT*, 'ENTER 'ID' (E.G. 'WA' OR 'DE')'	STYPES	188
77		REWIND 1	STYPES	189
78		READ(1,*,END=300) ID	STYPES	190
79		IF (((ICCHAR(ID(1:1)) .EQ. 55)	STYPES	191
80	+	.OR. (ICCHAR(ID(1:1)) .EQ. 36))	STYPES	192
81	+	.AND. (ICCHAR(ID(2:2)) .GE. 33)	STYPES	193
82	+	.AND. (ICCHAR(ID(2:2)) .LE. 58)	STYPES	194
83	+	.AND. (GETLEN(ID) .EQ. 2)) THEN	STYPES	195
84		TYPE(LINE,1) = ID	STYPES	196
85		ELSE	STYPES	197
86		PRINT*	STYPES	198
87		PRINT*, 'INCORRECT ENTRY. TRY AGAIN!!'	STYPES	199
88		GOTO 300	STYPES	200
89		END IF	STYPES	201
90	*		STYPES	202
91	440	PRINT*	STYPES	203
92		PRINT*, 'ENTER HEIGHT, METERS'	STYPES	204
93		REWIND 1	STYPES	205
94		READ(1,*,END=440) TDIM(LINE,1)	STYPES	206
95	*		STYPES	207
96	460	PRINT*	STYPES	208
97		PRINT*, 'ENTER WIDTH, METERS'	STYPES	209
98		REWIND 1	STYPES	210
99		READ(1,*,END=460) TDIM(LINE,2)	STYPES	211
100	*		STYPES	212
101	470	PRINT*	STYPES	213
102		PRINT*, 'ENTER DISTANCE ABOVE FLOOR, METERS'	STYPES	214
103		REWIND 1	STYPES	215
104		READ(1,*,END=470) TDIM(LINE,4)	STYPES	216
105	*		STYPES	217
106	480	PRINT*	STYPES	218
107		PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'	STYPES	219
108		REWIND 1	STYPES	220
109		READ(1,*,END=480) TDIM(LINE,3)	STYPES	221
110	*		STYPES	222
111	500	PRINT*	STYPES	223
112		PRINT*, 'ENTER "MATERIAL ID OF LAYER" (E.G. 'M01')'	STYPES	224
113		REWIND 1	STYPES	225
114		READ(1,*,END=500) MAT	STYPES	226
115		IF ((GETLEN(MAT) .EQ. 3)	STYPES	227
116	+	.AND. (MAT(1:1) .EQ. 'M')	STYPES	228
117	+	.AND. (ICCHAR(MAT(2:2)) .GE. 16)	STYPES	229
118	+	.AND. (ICCHAR(MAT(2:2)) .LE. 25)	STYPES	230
119	+	.AND. (ICCHAR(MAT(3:3)) .GE. 16)	STYPES	231
120	+	.AND. (ICCHAR(MAT(3:3)) .LE. 25)) THEN	STYPES	232
121		TYPE(LINE,2) = MAT	STYPES	233
122		ELSE	STYPES	234
123		PRINT*	STYPES	235
124		PRINT*, 'INCORRECT ENTRY!! TRY AGAIN'	STYPES	236
125		GOTO 500	STYPES	237
126		END IF	STYPES	238
127	*		STYPES	239
128	510	PRINT*	STYPES	240

129	PRINT*, 'ENTER "MATERIAL ID OF FRAME" (E.G. 'M01')'	STYPES	241
130	REWIND 1	STYPES	242
131	READ(1,*,END=510) FMAT	STYPES	243
132	IF ((GETLEN(FMAT).EQ. 3)	STYPES	244
133	+ .AND. (FMAT(1:1) .EQ. 'M')	STYPES	245
134	+ .AND. (ICHAR(FMAT(2:2)) .GE. 16)	STYPES	246
135	+ .AND. (ICHAR(FMAT(2:2)) .LE. 25)	STYPES	247
136	+ .AND. (ICHAR(FMAT(3:3)) .GE. 16)	STYPES	248
137	+ .AND. (ICHAR(FMAT(3:3)) .LE. 25)) THEN	STYPES	249
138	TYPE(LINE,3) = FMAT	STYPES	250
139	ELSE	STYPES	251
140	PRINT*	STYPES	252
141	PRINT*, 'INCORRECT ENTRY!! TRY AGAIN'	STYPES	253
142	GOTO 510	STYPES	254
143	END IF	STYPES	255
144	END IF	STYPES	256
145 *		STYPES	257
146 *		STYPES	258
147 *		STYPES	259
148	IF (ANSWER .EQ.2) THEN	STYPES	260
149 580	PRINT*	STYPES	261
150	PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'	STYPES	262
151	REWIND 1	STYPES	263
152	READ(1,*,END=580) TDIM(LINE,3)	STYPES	264
153 *		STYPES	265
154 600	PRINT*	STYPES	266
155	PRINT*, 'ENTER "MATERIAL ID OF LAYER" (E.G. 'M01')'	STYPES	267
156	REWIND 1	STYPES	268
157	READ(1,*,END=600) MAT	STYPES	269
158	IF ((GETLEN(MAT) .EQ. 3)	STYPES	270
159	+ .AND. (MAT(1:1) .EQ. 'M')	STYPES	271
160	+ .AND. (ICHAR(MAT(2:2)) .GE. 16)	STYPES	272
161	+ .AND. (ICHAR(MAT(2:2)) .LE. 25)	STYPES	273
162	+ .AND. (ICHAR(MAT(3:3)) .GE. 16)	STYPES	274
163	+ .AND. (ICHAR(MAT(3:3)) .LE. 25)) THEN	STYPES	275
164	TYPE(LINE,2) = MAT	STYPES	276
165	ELSE	STYPES	277
166	PRINT*	STYPES	278
167	PRINT*, 'INCORRECT ENTRY!! TRY AGAIN'	STYPES	279
168	GOTO 600	STYPES	280
169	END IF	STYPES	281
170	TYPE(LINE,3) = TYPE(LINE-1,3)	STYPES	282
171	TYPE(LINE,1) = TYPE(LINE-1,1)	STYPES	283
172	TDIM(LINE,1) = TDIM(LINE-1,1)	STYPES	284
173	TDIM(LINE,2) = TDIM(LINE-1,2)	STYPES	285
174	TDIM(LINE,4) = TDIM(LINE-1,4)	STYPES	286
175	END IF	STYPES	287
176	RETURN	STYPES	288
177	END	STYPES	289

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ABOVE	NONE	UNUSED/*S*	REAL
ANSWER	1076B		INTEGER
DIR	NONE	UNUSED/*S*	CHAR*3
DOK	NONE	UNUSED/*S*	INTEGER
FMAT	1101B		CHAR*3
H	NONE	UNUSED/*S*	REAL
ID	1077B		CHAR*3
INSERT	1	DUMMY-ARG	INTEGER
LINE	2	DUMMY-ARG	INTEGER
LOK	NONE	UNUSED/*S*	INTEGER

MAT	1100B		CHAR*3
NOK	NONE	UNUSED/*S*	INTEGER
NROOMS	1244B /ROOMN/		INTEGER
RAREA	1245B /ROOMN/		REAL 20
ROOM	0B /ROOMN/		REAL 676
T	NONE	UNUSED/*S*	REAL
TDBTOT	323B /TYPEN/		INTEGER
TDB1	37B /TYPEPEC/		CHAR*3 35
TDB2	215B /TYPEN/		REAL 70
TDIM	0B /TYPEN/		REAL 140
TERR	324B /TYPEN/		INTEGER
TTOT	214B /TYPEN/		INTEGER
TYPE	0B /TYPEPEC/		CHAR*3 105
V	NONE	UNUSED/*S*	INTEGER
VAL	NONE	UNUSED/*S*	INTEGER
W	NONE	UNUSED/*S*	REAL

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

RMAX	INTEGER	20
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

99	7B	45	480	213B	106
100	16B	49	500	227B	111
300	70B	75	510	311B	128
440	147B	91	580	377B	149
460	163B	96	600	413B	154
470	177B	101			

--ENTRY POINTS--(LO=A)

-NAME-----ADDRESS-----ARGS-----

DATAIN	5B	2
--------	----	---

--I/O UNITS--(LO=A)

-NAME-----PROPERTIES-----

TAPE1	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	1104B = 580
CM LABELLED COMMON LENGTH	1670B = 952
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.328 SECONDS

```

1      SUBROUTINE MANIP (QUIT,ABORT)                                STYPES      290
2      *****COMT 1
3      *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT 2
4      *****COMT 3
5      INTEGER TMAX COMT 4
6      PARAMETER (TMAX=35) COMT 5
7      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR COMT 6
8      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX) COMT 7
9      INTEGER TTOT,TDBTOT,TERR COMT 8
10     REAL TDIM,TDB2 COMT 9
11     CHARACTER * 3 TYPE,TDB1 COMT 10
12     *-----COMT 11
13     * DESCRIPTION OF ARRAYS COMT 12
14     *-----COMT 13
15     * ID MATERIAL FRAME MATERIAL COMT 14
16     *-----COMT 15
17     *TYPE(X,1) TYPE(X,2) TYPE(X,3) COMT 16
18     * A3 A3 A3 COMT 17
19     *-----COMT 18
20     * HEIGHT WIDTH LAYER DISTANCE COMT 19
21     * THICKNESS ABOVE FLOOR COMT 20
22     *-----COMT 21
23     * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4) COMT 22
24     * F8.2 F8.2 F8.2 F8.2 COMT 23
25     *-----COMT 24
26     * ID ATTENUATION AREA COMT 25
27     *-----COMT 26
28     * TDB1(X) TDB2(X,1) TDB2(X,2) COMT 27
29     * A3 E9.3 E9.3 COMT 28
30     *****COMT 29
31     *****COMT 30
32     INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT STYPES 292
33     INTEGER TEMP,V,X,Y,COMMAND STYPES 293
34     * STYPES 294
35 10 FLAG1 = 0 STYPES 295
36     PRINT* STYPES 296
37     PRINT*, 'CHOOSE' STYPES 297
38     PRINT*, ' (1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES' STYPES 298
39     PRINT*, ' (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA' STYPES 299
40     PRINT*, ' (3) DELETE LINE (6) STORE DATA AND EXIT ', STYPES 300
41     + 'PROGRAM' STYPES 301
42     PRINT*, ' (7) EXIT PROGRAM WITHOUT ', STYPES 302
43     + 'STORING DATA' STYPES 303
44     PRINT*, 'ENTER A NUMBER !!!' STYPES 304
45     PRINT* STYPES 305
46     REWIND 1 STYPES 306
47     READ(1,*,END=10) COMMAND STYPES 307
48     * STYPES 308
49     *----- STYPES 309
50     *** DISPLAY LINE *** STYPES 310
51     *----- STYPES 311
52     IF ( COMMAND .EQ. 1 ) THEN STYPES 312
53     * STYPES 313
54     *** INDICATE EMPTY DATA FILE STYPES 314
55     IF ( TTOT .EQ. 0 ) THEN STYPES 315
56     PRINT* STYPES 316
57     PRINT*, 'DATA FILE IS EMPTY !!!' STYPES 317
58     * STYPES 318
59     *** ENTER NUMBER OF LINE TO BE DISPLAYED STYPES 319
60     ELSE STYPES 320
61 100 PRINT* STYPES 321
62     PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED' STYPES 322
63     PRINT*, ' ( ENTER "0" TO ESCAPE DISPLAY MODE )' STYPES 323
64     REWIND 1 STYPES 324

```


65	READ(1,*,END=100) N	STYPES	325
66 *		STYPES	326
67 *** CHECK VALIDITY OF LINE NUMBER		STYPES	327
68	IF ((N .GT. TTOT) .OR. (N .LT. 0)) THEN	STYPES	328
69	PRINT*	STYPES	329
70	PRINT*, 'INCORRECT NUMBER !!!!! TRY AGAIN !!!'	STYPES	330
71	PRINT*, ' -OR- ENTER "0" TO ESCAPE ID ',	STYPES	331
72	+ "DISPLAY" MODE'	STYPES	332
73	GOTO 100	STYPES	333
74 *		STYPES	334
75 *** ABORT 'DISPLAY' MODE		STYPES	335
76	ELSE IF (N .EQ. 0) THEN	STYPES	336
77	PRINT*	STYPES	337
78	PRINT*, ' "DISPLAY" MODE ABORTED !!!'	STYPES	338
79 *		STYPES	339
80 *** DISPLAY LINE OF DATA		STYPES	340
81	ELSE IF ((N .GT. 0) .AND. (N .LE. TTOT)) THEN	STYPES	341
82	PRINT*	STYPES	342
83	CALL DISPLAY(N, COMMAND)	STYPES	343
84 *		STYPES	344
85	END IF	STYPES	345
86	END IF	STYPES	346
87	END IF	STYPES	347
88 *		STYPES	348
89	-----	STYPES	349
90 *** INSERT LINE ***		STYPES	350
91	-----	STYPES	351
92	IF (COMMAND .EQ. 2) THEN	STYPES	352
93 *		STYPES	353
94 *** INDICATE EMPTY DATA FILE		STYPES	354
95	IF (TTOT .EQ. 0) THEN	STYPES	355
96	PRINT*	STYPES	356
97	PRINT*, 'DATA FILE IS EMPTY !!!'	STYPES	357
98 *		STYPES	358
99 *** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE		STYPES	359
100	ELSE	STYPES	360
101 200	PRINT*	STYPES	361
102	PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',	STYPES	362
103	+ 'TO BE INSERTED'	STYPES	363
104	PRINT*, ' (ENTER "0" TO ESCAPE "INSERTION" MODE)'	STYPES	364
105	REWIND 1	STYPES	365
106	READ(1,*,END=200) N	STYPES	366
107 *		STYPES	367
108 *** CHECK FOR VALID LINE NUMBER		STYPES	368
109	IF ((N .LT. 0) .OR. (N .GT. TTOT)) THEN	STYPES	369
110	PRINT*	STYPES	370
111	PRINT*, 'INCORRECT NUMBER !!!'	STYPES	371
112	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE',	STYPES	372
113	+ "INSERTION" MODE'	STYPES	373
114	GOTO 200	STYPES	374
115 *		STYPES	375
116 *** ABORT INSERTION MODE		STYPES	376
117	ELSE IF (N .EQ. 0) THEN	STYPES	377
118	PRINT*	STYPES	378
119	PRINT*, ' "INSERTION" MODE ABORTED'	STYPES	379
120 *		STYPES	380
121 *** MAKE ROOM FOR NEW LINE OF DATA		STYPES	381
122	ELSE IF ((N .GT. 0) .AND. (N .LE. TTOT)) THEN	STYPES	382
123	DO 230 X = TTOT, N, -1	STYPES	383
124	DO 210 Y = 1, 3	STYPES	384
125	TYPE(X+1, Y) = TYPE(X, Y)	STYPES	385
126 210	CONTINUE	STYPES	386
127	DO 220 Y = 1, 4	STYPES	387
128	TDIM(X+1, Y) = TDIM(X, Y)	STYPES	388

129	220	CONTINUE	STYPES	389
130	230	CONTINUE	STYPES	390
131	*		STYPES	391
132	***	ENTER DATA FOR NEW LINE	STYPES	392
133		TTOT = TTOT + 1	STYPES	393
134		CALL DATAIN (1,N)	STYPES	394
135	*		STYPES	395
136	***	INITIALIZE FLAGS	STYPES	396
137		OK1 = 0	STYPES	397
138		OK2 = 0	STYPES	398
139		OK = 0	STYPES	399
140	*		STYPES	400
141	***	TEST VALIDITY OF DATA	STYPES	401
142	*		STYPES	402
143	***	TEST IF NEW LAYER BELONGS TO THE NEXT TYPE	STYPES	403
144		IF ((TYPE(N,1) .EQ. TYPE(N+1,1))	STYPES	404
145	+	.AND. (TYPE(N,3) .EQ. TYPE(N+1,3))) THEN	STYPES	405
146		IF ((TDIM(N,1) .EQ. TDIM(N+1,1))	STYPES	406
147	+	.AND. (TDIM(N,2) .EQ. TDIM(N+1,2))	STYPES	407
148	+	.AND. (TDIM(N,4) .EQ. TDIM(N+1,4))) THEN	STYPES	408
149		OK1 = 1	STYPES	409
150		END IF	STYPES	410
151		END IF	STYPES	411
152	*		STYPES	412
153	***	TEST IF NEW LAYER BELONGS TO PREVIOUS TYPE	STYPES	413
154		IF (N .GT. 1) THEN	STYPES	414
155		IF ((TYPE(N,1) .EQ. TYPE(N-1,1))	STYPES	415
156	+	.AND. (TYPE(N,3) .EQ. TYPE(N-1,3))) THEN	STYPES	416
157		IF ((TDIM(N,1) .EQ. TDIM(N-1,1))	STYPES	417
158	+	.AND. (TDIM(N,2) .EQ. TDIM(N-1,2))	STYPES	418
159	+	.AND. (TDIM(N,4) .EQ. TDIM(N-1,4))) THEN	STYPES	419
160		OK2 = 1	STYPES	420
161		END IF	STYPES	421
162		END IF	STYPES	422
163		END IF	STYPES	423
164	*		STYPES	424
165		IF ((OK1 .EQ. 1) .OR. (OK2 .EQ. 1)) THEN	STYPES	425
166		OK = 1	STYPES	426
167		END IF	STYPES	427
168	*		STYPES	428
169		IF (OK .EQ. 1) THEN	STYPES	429
170		PRINT*	STYPES	430
171		PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N	STYPES	431
172		CALL DISPLAY(N, COMMAND)	STYPES	432
173	*		STYPES	433
174	***	REJECT DATA IF DATA DOESN'T MATCH PREVIOUS OR NEXT LAYER	STYPES	434
175		ELSE IF (OK .EQ. 0) THEN	STYPES	435
176		PRINT*	STYPES	436
177		PRINT*, 'YOUR DATA WAS NOT ACCEPTED !!!'	STYPES	437
178		PRINT*, 'YOUR DATA MUST REPRESENT A LAYER ',	STYPES	438
179	+	'IN AN EXISTING DOOR OR WINDOW'	STYPES	439
180		PRINT*, ' I.E. THE ID, FRAME MATERIAL, HEIGHT, ',	STYPES	440
181	+	'WIDTH, AND DISTANCE ABOVE FLOOR'	STYPES	441
182		PRINT*, ' PARAMETERS MUST MATCH THE DOOR OR ',	STYPES	442
183	+	'WINDOW JUST BEFORE'	STYPES	443
184		PRINT*, ' OR JUST AFTER YOUR SPECIFIED INSERTION ',	STYPES	444
185	+	'POINT'	STYPES	445
186		PRINT*	STYPES	446
187		PRINT*, 'THE FOLLOWING DISPLAYS'	STYPES	447
188		IF (N .GT. 1) PRINT*, 'THE LINE BEFORE YOUR LINE, '	STYPES	448
189		PRINT*, 'YOUR LINE, AND THE LINE AFTER'	STYPES	449
190		PRINT*	STYPES	450
191	*		STYPES	451
192	***	DISPLAY LINES OF DATA	STYPES	452

193	IF (N .GT. 1) CALL DISPLAY (N-1, COMMAND)	STYPES	453
194	CALL DISPLAY(N, COMMAND)	STYPES	454
195	CALL DISPLAY (N+1, COMMAND)	STYPES	455
196 *		STYPES	456
197 *** REMOVE THE LINE OF INCORRECTLY ENTERED DATA		STYPES	457
198	DO 270 X = N,TTOT	STYPES	458
199	DO 250 Y = 1,3	STYPES	459
200	TYPE(X,Y) = TYPE(X+1,Y)	STYPES	460
201 250	CONTINUE	STYPES	461
202	DO 260 Y = 1,4	STYPES	462
203	TDIM(X,Y) = TDIM(X+1,Y)	STYPES	463
204 260	CONTINUE	STYPES	464
205 270	CONTINUE	STYPES	465
206	TTOT = TTOT - 1	STYPES	466
207	END IF	STYPES	467
208	END IF	STYPES	468
209	END IF	STYPES	469
210	END IF	STYPES	470
211 *		STYPES	471
212 *	-----	STYPES	472
213 *** DELETE LINE ***		STYPES	473
214 *	-----	STYPES	474
215	IF (COMMAND .EQ. 3) THEN	STYPES	475
216 *		STYPES	476
217 *** INDICATE EMPTY DATA FILE		STYPES	477
218	IF (TTOT .EQ. 0) THEN	STYPES	478
219	PRINT*	STYPES	479
220	PRINT*, 'DATA FILE IS EMPTY !!!'	STYPES	480
221 *		STYPES	481
222 *** READ NUMBER OF LINE TO BE DELETED		STYPES	482
223	ELSE	STYPES	483
224 300	PRINT*	STYPES	484
225	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'	STYPES	485
226	PRINT*, ' (ENTER "0" TO ESCAPE DELETION MODE)'	STYPES	486
227	REWIND 1	STYPES	487
228	READ(1,*,END=300) N	STYPES	488
229 *		STYPES	489
230 *** CHECK VALIDITY OF LINE NUMBER		STYPES	490
231	IF ((N .GT. TTOT) .OR. (N .LT. 0)) THEN	STYPES	491
232	PRINT*	STYPES	492
233	PRINT*, ' INCORRECT NUMBER !!!'	STYPES	493
234	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE ID',	STYPES	494
235	+ "DELETE" MODE'	STYPES	495
236	GOTO 300	STYPES	496
237 *		STYPES	497
238 *** ABORT 'DELETE' MODE		STYPES	498
239	ELSE IF (N .EQ. 0) THEN	STYPES	499
240	PRINT*, ' "DELETE" MODE ABORTED'	STYPES	500
241 *		STYPES	501
242 *** DOUBLE CHECK CHOICE OF LINE TO BE DELETED		STYPES	502
243	ELSE IF ((N .GT. 0) .AND. (N .LE. TTOT)) THEN	STYPES	503
244	PRINT*	STYPES	504
245	PRINT*, 'DOUBLE CHECK !!!'	STYPES	505
246	PRINT*, ' DO YOU WANT TO DELETE THE FOLLOWING LINE?:'	STYPES	506
247	CALL DISPLAY(N, COMMAND)	STYPES	507
248 305	PRINT*, ' ENTER (1) YES OR (2) NO'	STYPES	508
249	REWIND 1	STYPES	509
250	READ(1,*,END=305) ANSWER	STYPES	510
251 *		STYPES	511
252 *** DELETE LINE		STYPES	512
253	IF (ANSWER .EQ. 1) THEN	STYPES	513
254	DO 330 X = N, TTOT - 1	STYPES	514
255	DO 310 Y = 1,3	STYPES	515
256	TYPE(X,Y) = TYPE(X+1,Y)	STYPES	516

257 310	CONTINUE	STYPES	517
258	DO 320 Y = 1,4	STYPES	518
259	TDIM(X,Y) = TDIM(X+1,Y)	STYPES	519
260 320	CONTINUE	STYPES	520
261 330	CONTINUE	STYPES	521
262	TTOT = TTOT - 1	STYPES	522
263	PRINT*	STYPES	523
264	PRINT*, 'LINE # ',N,' DELETED'	STYPES	524
265	END IF	STYPES	525
266 *		STYPES	526
267	END IF	STYPES	527
268	END IF	STYPES	528
269	END IF	STYPES	529
270 *		STYPES	530
271 *	-----	STYPES	531
272 ***	DISPLAY ALL DATA ***	STYPES	532
273 *	-----	STYPES	533
274	IF (COMMAND .EQ. 4) THEN	STYPES	534
275 *		STYPES	535
276 ***	INDICATE EMPTY DATA FILE	STYPES	536
277	IF (TTOT .EQ. 0) THEN	STYPES	537
278	PRINT*	STYPES	538
279	PRINT*, 'DATA FILE IS EMPTY !!!'	STYPES	539
280 *		STYPES	540
281 ***	DISPLAY DATA	STYPES	541
282	ELSE	STYPES	542
283	PRINT*	STYPES	543
284	CALL DISPLAY(N, COMMAND)	STYPES	544
285 *		STYPES	545
286	END IF	STYPES	546
287	END IF	STYPES	547
288 *		STYPES	548
289 *	-----	STYPES	549
290 ***	ADD DATA ***	STYPES	550
291 *	-----	STYPES	551
292	IF (COMMAND .EQ. 5) THEN	STYPES	552
293 *		STYPES	553
294 ***	ENTER DATA	STYPES	554
295 500	TTOT = TTOT + 1	STYPES	555
296 *		STYPES	556
297	CALL DATAIN (0,TTOT)	STYPES	557
298 510	PRINT*	STYPES	558
299	PRINT*, 'DO YOU WANT TO ENTER MORE DATA? (1) YES (2) NO'	STYPES	559
300	PRINT*, ' ENTER A NUMBER !!!'	STYPES	560
301	REWIND 1	STYPES	561
302	READ(1,*,END=510) ANSWER	STYPES	562
303 *		STYPES	563
304 ***	CHECK VALIDITY OF NUMBER	STYPES	564
305 *		STYPES	565
306 ***	ENTER MORE DATA	STYPES	566
307	IF (ANSWER .EQ. 1) THEN	STYPES	567
308	GOTO 500	STYPES	568
309 *		STYPES	569
310 ***	DISCONTINUE DATA ENTRY	STYPES	570
311	ELSE IF (ANSWER .EQ. 2) THEN	STYPES	571
312	PRINT*	STYPES	572
313	PRINT*, 'DATA ENTRY DISCONTINUED'	STYPES	573
314 *		STYPES	574
315 ***	INVALID ENTRY	STYPES	575
316	ELSE	STYPES	576
317	GOTO 510	STYPES	577
318	END IF	STYPES	578
319	END IF	STYPES	579
320 *		STYPES	580

321 *	-----	STYPES	581
322 *** STORE DATA AND PROGRAM ***		STYPES	582
323 *	-----	STYPES	583
324 IF (COMMAND .EQ. 6) THEN		STYPES	584
325 600 PRINT*		STYPES	585
326 PRINT*, 'DOUBLE CHECK !!!'		STYPES	586
327 PRINT*, ' DO YOU YOU WANT TO STORE THIS DATA AND END PROG'		STYPES	587
328 PRINT*, ' NOTE. STORING THIS DATA WILL WIPE OUT ANY OLD FILE '		STYPES	588
329 PRINT*, ' OF THE SAME NAME !!!'		STYPES	589
330 PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'		STYPES	590
331 REWIND 1		STYPES	591
332 READ(1,*,END=600) ANSWER		STYPES	592
333 *		STYPES	593
334 *** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM		STYPES	594
335 IF (ANSWER .EQ. 1) THEN		STYPES	595
336 QUIT = 1		STYPES	596
337 RETURN		STYPES	597
338 *		STYPES	598
339 *** ABORT 'STORING' MODE		STYPES	599
340 ELSE IF (ANSWER .EQ. 2) THEN		STYPES	600
341 PRINT*		STYPES	601
342 PRINT*, ' "STORING" MODE DISCONTINUED'		STYPES	602
343 *		STYPES	603
344 *** CHECK VALIDITY OF ANSWER		STYPES	604
345 ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN		STYPES	605
346 GOTO 600		STYPES	606
347 *		STYPES	607
348 END IF		STYPES	608
349 END IF		STYPES	609
350 *		STYPES	610
351 *	-----	STYPES	611
352 *** END PROGRAM WITHOUT STORING DATA ***		STYPES	612
353 *	-----	STYPES	613
354 IF (COMMAND .EQ. 7) THEN		STYPES	614
355 700 PRINT*		STYPES	615
356 PRINT*, 'DOUBLE CHECK !!!'		STYPES	616
357 PRINT*, ' DO YOU WANT TO END THIS PROGRAM ',		STYPES	617
358 + 'WITHOUT STORING DATA?'		STYPES	618
359 PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'		STYPES	619
360 REWIND 1		STYPES	620
361 READ(1,*,END=700) ANSWER		STYPES	621
362 *		STYPES	622
363 *** SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM		STYPES	623
364 IF (ANSWER .EQ. 1) THEN		STYPES	624
365 ABORT = 1		STYPES	625
366 RETURN		STYPES	626
367 *		STYPES	627
368 *** ABORT 'STORING' MODE		STYPES	628
369 ELSE IF (ANSWER .EQ. 2) THEN		STYPES	629
370 PRINT*		STYPES	630
371 PRINT*, ' "ABORTION" MODE DISCONTINUED'		STYPES	631
372 *		STYPES	632
373 *** CHECK VALIDITY OF ANSWER		STYPES	633
374 ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN		STYPES	634
375 GOTO 700		STYPES	635
376 *		STYPES	636
377 END IF		STYPES	637
378 END IF		STYPES	638
379 *		STYPES	639
380 *	-----	STYPES	640
381 *** LOOP TO BEGINNING OF 'MANIP' SUBROUTINE		STYPES	641
382 *	-----	STYPES	642
383 GOTO 10		STYPES	643
384 *		STYPES	644

385 RETURN
 TRIVIAL* NO PATH TO THIS STATEMENT
 386 END

STYPES 645
 STYPES 646

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	2	DUMMY-ARG	INTEGER	
ANSWER	2030B		INTEGER	
COMMAND	2040B		INTEGER	
DOK	NONE	UNUSED/*S*	INTEGER	
FLAG1	2031B		INTEGER	
INSERT	NONE	UNUSED/*S*	INTEGER	
LOK	NONE	UNUSED/*S*	INTEGER	
N	2032B		INTEGER	
NOK	NONE	UNUSED/*S*	INTEGER	
OK	2033B		INTEGER	
OK1	2034B		INTEGER	
OK2	2035B		INTEGER	
QUIT	1	DUMMY-ARG	INTEGER	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPECL/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TEMP	NONE	UNUSED/*S*	INTEGER	
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPECL/	CHAR*3	105
V	NONE	UNUSED/*S*	INTEGER	
X	2036B		INTEGER	
Y	2037B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

TMAX	INTEGER	35
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

DATAIN	2	SUBROUTINE
DISPLAY	2	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	7B	35	300	547B	224
100	50B	61	305	616B	248
200	133B	101	310	INACTIVE	DO-TERM 257
210	INACTIVE	DO-TERM 126	320	INACTIVE	DO-TERM 260
220	INACTIVE	DO-TERM 129	330	INACTIVE	DO-TERM 261
230	INACTIVE	DO-TERM 130	500	741B	295
250	INACTIVE	DO-TERM 201	510	745B	298
260	INACTIVE	DO-TERM 204	600	1001B	325
270	INACTIVE	DO-TERM 205	700	1053B	355

FTN 5.1+552 83/12/24. 10.36.12 PAGE 17
SUBROUTINE MANIP 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

MANIP 5B 2

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE: FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	2054B = 1068
CM LABELLED COMMON LENGTH	377B = 255
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.553 SECONDS

1 TRIVIAL ERROR IN MANIP

```

1      SUBROUTINE DISPLAY (LINE, COMMAND)                                STYPES 647
2      *****COMT 1
3      *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT 2
4      *****COMT 3
5      INTEGER TMAX                                                        COMT 4
6      PARAMETER (TMAX=35)                                                COMT 5
7      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT 6
8      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT 7
9      INTEGER TTOT,TDBTOT,TERR                                           COMT 8
10     REAL TDIM,TDB2                                                      COMT 9
11     CHARACTER * 3 TYPE,TDB1                                           COMT 10
12     *=====COMT 11
13     * DESCRIPTION OF ARRAYS                                             COMT 12
14     *=====COMT 13
15     * ID MATERIAL FRAME MATERIAL COMT 14
16     *-----COMT 15
17     *TYPE(X,1) TYPE(X,2) TYPE(X,3) COMT 16
18     * A3 A3 A3 COMT 17
19     *=====COMT 18
20     * HEIGHT WIDTH LAYER DISTANCE COMT 19
21     * THICKNESS ABOVE FLOOR COMT 20
22     *-----COMT 21
23     * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4) COMT 22
24     * F8.2 F8.2 F8.2 F8.2 COMT 23
25     *=====COMT 24
26     * ID ATTENUATION AREA COMT 25
27     *-----COMT 26
28     * TDB1(X) TDB2(X,1) TDB2(X,2) COMT 27
29     * A3 E9.3 E9.3 COMT 28
30     *****COMT 29
31     *****COMT 30
32     INTEGER LINE, COMMAND, N STYPES 649
33     1000 FORMAT (1X,A5,2X,A2,2X,A8,2X,A8,2X,A11,2X,A9,2X,A8,2X,A8) STYPES 650
34     2000 FORMAT (2X,I3,3X,A2,3X,F6.2,4X,F6.2,4X,F6.2,7X,F6.2,6X,A3, STYPES 651
35     + 7X,A3) STYPES 652
36     PRINT 1000, 'LINE','ID','HEIGHT ','WIDTH ','DISTANCE ', STYPES 653
37     + 'THICKNESS','LAYER ','FRAME ' STYPES 654
38     PRINT 1000, ' ',' ','(METERS)','(METERS)','ABOVE FLOOR', STYPES 655
39     + '(CM) ','MATERIAL','MATERIAL' STYPES 656
40     IF ( COMMAND .EQ. 4 ) THEN STYPES 657
41     DO 10 LINE = 1,TTOT STYPES 658
42     PRINT 2000, LINE,TYPE(LINE,1),TDIM(LINE,1),TDIM(LINE,2), STYPES 659
43     + TDIM(LINE,4),TDIM(LINE,3),TYPE(LINE,2),TYPE(LINE,3) STYPES 660
44 10 CONTINUE STYPES 661
45     ELSE STYPES 662
46     PRINT 2000, LINE,TYPE(LINE,1),TDIM(LINE,1),TDIM(LINE,2), STYPES 663
47     + TDIM(LINE,4),TDIM(LINE,3),TYPE(LINE,2),TYPE(LINE,3) STYPES 664
48     END IF STYPES 665
49     RETURN STYPES 666
50     END STYPES 667

```

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

COMMAND	2	DUMMY-ARG	INTEGER	
LINE	1	DUMMY-ARG	INTEGER	
N	NONE	UNUSED/*S*	INTEGER	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPEC/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	

TYPE 0B /TYPEC/ CHAR*3 105

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

TMAX INTEGER 35

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES---DEF

10	INACTIVE	DO-TERM	44
1000	163B	FORMAT	33
2000	171B	FORMAT	34

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

DISPLAY 5B 2

--STATISTICS--

PROGRAM-UNIT LENGTH	254B = 172
CM LABELLED COMMON LENGTH	377B = 255
CM STORAGE USED	61000B = 25088
COMPILE TIME	0 100 SECONDS

1	INTEGER FUNCTION VAL(String)	STYPES	668
2	C** RETURNS THE INTEGER VALUE OF A STRING.	STYPES	669
3	INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN	STYPES	670
4	CHARACTER * (*) STRING	STYPES	671
5	L = GETLEN(String)	STYPES	672
6	NUMBER = 0	STYPES	673
7	DO 10 X = L,1,-1	STYPES	674
8	EXP = L - X	STYPES	675
9	DIGIT = ICHAR(String(X:X)) - 16	STYPES	676
10	NUMBER = NUMBER + DIGIT*10**EXP	STYPES	677
11	10 CONTINUE	STYPES	678
12	VAL = NUMBER	STYPES	679
13	RETURN	STYPES	680
14	END	STYPES	681

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK----PROPERTIES-----TYPE-----SIZE

DIGIT	76B	INTEGER
EXP	75B	INTEGER
L	74B	INTEGER
NUMBER	72B	INTEGER
STRING	1 DUMMY-ARG	CHAR*(*)
VAL	71B	INTEGER
X	73B	INTEGER

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES---DEF

10	INACTIVE	DO-TERM	11
----	----------	---------	----

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

VAL	6B	1
-----	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.041 SECONDS

1	INTEGER FUNCTION GETLEN (STRING)	STYPES	682
2	C	STYPES	683
3	C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING	STYPES	684
4	C	STYPES	685
5	C	STYPES	686
6	C ARGUMENT DEFINITIONS --	STYPES	687
7	C READ ARGUMENTS	STYPES	688
8	C STRING - STRING WHOSE LENGTH IS TO BE DETERMINED	STYPES	689
9	C	STYPES	690
10	CHARACTER * (*) STRING	STYPES	691
11	C	STYPES	692
12	C FUNCTION PARAMETERS	STYPES	693
13	CHARACTER * 1 BLANK	STYPES	694
14	PARAMETER (BLANK = ' ')	STYPES	695
15	C	STYPES	696
16	C LOCAL VARIABLES	STYPES	697
17	INTEGER NEXT	STYPES	698
18	C	STYPES	699
19	C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK	STYPES	700
20	DO 10 NEXT = LEN(STRING),1,-1	STYPES	701
21	IF (STRING(NEXT : NEXT) .NE. BLANK) THEN	STYPES	702
22	GETLEN = NEXT	STYPES	703
23	RETURN	STYPES	704
24	END IF	STYPES	705
25	10 CONTINUE	STYPES	706
26	C	STYPES	707
27	C ALL CHARACTERS ARE BLANKS	STYPES	708
28	GETLEN = 0	STYPES	709
29	C	STYPES	710
30	RETURN	STYPES	711
31	END	STYPES	712

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

GETLEN	63B		INTEGER
NEXT	64B		INTEGER
STRING	1	DUMMY-ARG	CHAR*(*)

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME---TYPE-----VALUE

BLANK	CHAR*1	' '
-------	--------	-----

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

LEN	INTEGER	1	INTRINSIC
-----	---------	---	-----------

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	25
----	----------	---------	----

FTN 5.1+552 83/12/24. 10.36.12 PAGE 22
 FUNCTION GETLEN 74/175 OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS---ARGS---

GETLEN	6B	1
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	70B = 56
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.041 SECONDS

```

1      SUBROUTINE LTYPE                                LTYPE      1
2      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LTYPE      2
3      *!!!!                                           !!!LTYPE      3
4      *!!!!   LOAD THE "TYPE" ARRAYS FROM THE TYPE DATA FILE   !!!LTYPE      4
5      *!!!!                                           !!!LTYPE      5
6      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LTYPE      6
7      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LTYPE      7
8      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! COMT      1
9      ***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS   ***COMT      2
10     *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! COMT      3
11     INTEGER TMAX                                           COMT      4
12     PARAMETER (TMAX=35)                                     COMT      5
13     COMMON /TYPE/ TDIM(TMAX,4), TTOT, TDB2(TMAX,2), TDBTOT, TERR COMT      6
14     COMMON /TYPE/ TYPE(TMAX,3), TDB1(TMAX)                 COMT      7
15     INTEGER TTOT, TDBTOT, TERR                             COMT      8
16     REAL TDIM, TDB2                                         COMT      9
17     CHARACTER * 3 TYPE, TDB1                              COMT     10
18     *=====                                              COMT     11
19     * DESCRIPTION OF ARRAYS                                COMT     12
20     *=====                                              COMT     13
21     *   ID          MATERIAL      FRAME MATERIAL          COMT     14
22     *-----                                              COMT     15
23     *TYPE(X,1)      TYPE(X,2)      TYPE(X,3)              COMT     16
24     *   A3          A3             A3                     COMT     17
25     *=====                                              COMT     18
26     * HEIGHT        WIDTH          LAYER      DISTANCE    COMT     19
27     *               THICKNESS     ABOVE FLOOR            COMT     20
28     *-----                                              COMT     21
29     * TDIM(X,1)      TDIM(X,2)      TDIM(X,3)  TDIM(X,4)  COMT     22
30     *   F8.2         F8.2           F8.2       F8.2       COMT     23
31     *=====                                              COMT     24
32     *   ID          ATTENUATION    AREA              COMT     25
33     * -----                                              COMT     26
34     * TDB1(X)        TDB2(X,1)      TDB2(X,2)          COMT     27
35     *   A3           E9.3           E9.3              COMT     28
36     *=====                                              COMT     29
37     *=====                                              COMT     30
38     *=====                                              COMF      1
39     *** COMMON FOR INITIAL PARAMETERS                      ***COMF      2
40     *=====                                              COMF      3
41     INTEGER FMAX                                           COMF      4
42     PARAMETER (FMAX = 50)                                   COMF      5
43     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
44     $              FTOT                                     COMF      7
45     COMMON /INITILC/ BLDG                                     COMF      8
46     CHARACTER * 5 BLDG                                     COMF      9
47     REAL FREQ, AFLAG, RFLAG, FREQA                         COMF     10
48     INTEGER QUALITY, FERR, FTOT                             COMF     11
49     *=====                                              COMF     12
50     *=====                                              COMF     13
51     *=====                                              LTYPE     10
52     *   DECLARATION OF VARIABLES                          LTYPE     11
53     *=====                                              LTYPE     12
54     INTEGER GETLEN, R, C                                    LTYPE     13
55     CHARACTER * 7 PFN                                       LTYPE     14
56     *=====                                              LTYPE     15
57     *                                                     LTYPE     16
58     *=====                                              LTYPE     17
59     PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'T'             LTYPE     18
60     TERR = 0                                                LTYPE     19
61     CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',TERR)         LTYPE     20
62     IF (TERR .EQ. 0 ) THEN                                  LTYPE     21
63         OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',          LTYPE     22
64         $      STATUS='OLD', ACCESS='SEQUENTIAL')          LTYPE     23

```

65	1000	FORMAT (1X,3(1X,A3),4(1X,F8.2))	LTYPE	24
66		TTOT = 0	LTYPE	25
67		DO 10 R = 1,TMAX	LTYPE	26
68		READ (3,1000,END=20)(TYPE(R,C),C=1,3),(TDIM(R,C),C=1,4)	LTYPE	27
69		TTOT = TTOT + 1	LTYPE	28
70	10	CONTINUE	LTYPE	29
71	20	CONTINUE	LTYPE	30
72		CLOSE(3,STATUS='DELETE')	LTYPE	31
73		ELSE IF (TERR .EQ. 2) THEN	LTYPE	32
74		CALL WARNING (5)	LTYPE	33
75		ELSE	LTYPE	34
76		CALL WARNING (6)	LTYPE	35
77		END IF	LTYPE	36
78		RETURN	LTYPE	37
79		END	LTYPE	38

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	236B		INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
PFN	237B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
R	235B		INTEGER	
RFLAG	3B	/INITILN/	REAL	
TDBTOT	323B	/TYPE/	INTEGER	
TDB1	37B	/TYPE/	CHAR*3	35
TDB2	215B	/TYPE/	REAL	70
TDIM	0B	/TYPE/	REAL	140
TERR	324B	/TYPE/	INTEGER	
TTOT	214B	/TYPE/	INTEGER	
TYPE	0B	/TYPE/	CHAR*3	105

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	70
20	111B		71
1000	147B	FORMAT	65

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

LTYPE	5B	0
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE3 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	245B = 165
CM LABELLED COMMON LENGTH	470B = 312
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.107 SECONDS

1	SUBROUTINE ERROR(IERR)	ERROR	1
2	CHARACTER*45 MESSAGE(20)	ERROR	2
3	DATA MESSAGE(1) //'MATERIALS DATA BASE IS EMPTY	' / ERROR	3
4	DATA MESSAGE(2) //'FREQUENCY IS OUT OF RANGE	' / ERROR	4
5	DATA MESSAGE(3) //'THIS MATERIAL IS NOT IN DATA BASE	' / ERROR	5
6	DATA MESSAGE(4) //'DENOMINATOR IS ZERO	' / ERROR	6
7	DATA MESSAGE(5) //'FILE HANDLING ERROR	' / ERROR	7
8	DATA MESSAGE(6) //'ERROR CODE IS OUT OF RANGE	' / ERROR	8
9	DATA MESSAGE(7) //'ERROR CODE IS OUT OF RANGE	' / ERROR	9
10	DATA MESSAGE(8) //'ERROR CODE IS OUT OF RANGE	' / ERROR	10
11	DATA MESSAGE(9) //'ERROR CODE IS OUT OF RANGE	' / ERROR	11
12	DATA MESSAGE(10) //'ERROR CODE IS OUT OF RANGE	' / ERROR	12
13	DATA MESSAGE(11) //'ERROR CODE IS OUT OF RANGE	' / ERROR	13
14	DATA MESSAGE(12) //'ERROR CODE IS OUT OF RANGE	' / ERROR	14
15	DATA MESSAGE(13) //'ERROR CODE IS OUT OF RANGE	' / ERROR	15
16	DATA MESSAGE(14) //'ERROR CODE IS OUT OF RANGE	' / ERROR	16
17	DATA MESSAGE(15) //'ERROR CODE IS OUT OF RANGE	' / ERROR	17
18	DATA MESSAGE(16) //'ERROR CODE IS OUT OF RANGE	' / ERROR	18
19	DATA MESSAGE(17) //'ERROR CODE IS OUT OF RANGE	' / ERROR	19
20	DATA MESSAGE(18) //'ERROR CODE IS OUT OF RANGE	' / ERROR	20
21	DATA MESSAGE(19) //'ERROR CODE IS OUT OF RANGE	' / ERROR	21
22	DATA MESSAGE(20) //'ERROR CODE IS OUT OF RANGE	' / ERROR	22
23	IERRM=5	ERROR	23
24	IF(IERR.GT.IERRM) IERR=20	ERROR	24
25	WRITE(6,10) IERR,MESSAGE(IERR)	ERROR	25
26	10 FORMAT(' ***ERROR NUMBER = ',I5,' *** ',A45)	ERROR	26
27	CALL PMDSTOP	ERROR	27
28	STOP 'ERROR'	ERROR	28
29	END	ERROR	29

--VARIABLE MAP--(LO=A)
--NAME---ADDRESS--BLOCK----PROPERTIES-----TYPE-----SIZE

IERR	1	DUMMY-ARG	INTEGER	
IERRM	210B		INTEGER	
MESSAGE	56B		CHAR*45	20

--PROCEDURES--(LO=A)
--NAME-----TYPE-----ARGS-----CLASS-----

PMDSTOP	0	SUBROUTINE
---------	---	------------

--STATEMENT LABELS--(LO=A)
--LABEL-ADDRESS-----PROPERTIES-----DEF

10	36B	FORMAT	26
----	-----	--------	----

--ENTRY POINTS--(LO=A)
--NAME---ADDRESS--ARGS---

ERROR	5B	1
-------	----	---

--I/O UNITS--(LO=A)
--NAME--- PROPERTIES-----

TAPE6	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	213B = 139
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.054 SECONDS

1	SUBROUTINE WARNING(ERR)	WARNING	1
2	INTEGER ERR, ERRM	WARNING	2
3	CHARACTER*45 MESSAGE(20)	WARNING	3
4	DATA MESSAGE(1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	4
5	DATA MESSAGE(2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE	WARNING	5
6	DATA MESSAGE(3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG	WARNING	6
7	DATA MESSAGE(4)/'FILE HANDLING PROBLEM ON "MATTER" FILE	WARNING	7
8	DATA MESSAGE(5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	8
9	DATA MESSAGE(6)/'FILE HANDLING PROBLEM ON "TYPE" FILE	WARNING	9
10	DATA MESSAGE(7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'	WARNING	10
11	DATA MESSAGE(8)/'FILE HANDLING PROBLEM ON "WALL" FILE	WARNING	11
12	DATA MESSAGE(9)/'HEIGHT AND WIDTH OF ROOM MISSING	WARNING	12
13	DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING	WARNING	13
14	DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG	WARNING	14
15	DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE	WARNING	15
16	DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE	WARNING	16
17	DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE	WARNING	17
18	DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE	WARNING	18
19	DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE	WARNING	19
20	DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE	WARNING	20
21	DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE	WARNING	21
22	DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE	WARNING	22
23	DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE	WARNING	23
24	ERRM=12	WARNING	24
25	IERR = ERR	WARNING	25
26	IF(ERR.GT.ERRM) IERR=20	WARNING	26
27	WRITE(6,20)	WARNING	27
28	WRITE(6,10) ERR,MESSAGE(IERR)	WARNING	28
29	WRITE(6,20)	WARNING	29
30 10	FORMAT(' ***WARNING NUMBER = ',I5,' *** ',A45)	WARNING	30
31 20	FORMAT(' ')	WARNING	31
32	RETURN	WARNING	32
33	END	WARNING	33

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ERR	1	DUMMY-ARG	INTEGER
ERRM	60B		INTEGER
IERR	213B		INTEGER
MESSAGE	61B		CHAR*45 20

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	34B	FORMAT	30
20	42B	FORMAT	31

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARCS---

WARNING	5B	1
---------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	216B = 142
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.058 SECONDS

Appendix 9.6 Listing of Computer Program SFREQ

1	PROGRAM SFREQ (INPUT,TAPE1=INPUT)	SFREQ	1
2	*	SFREQ	2
3	*THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH FREQ	SFREQ	3
4	*IN THE BUILDING AND STORES IT. THE FILE NAME IS CREATED BY	SFREQ	4
5	*ATTACHING "B" TO THE FRONT OF AND "F" TO THE BACK OF THE BUILDING	SFREQ	5
6	*IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE	SFREQ	6
7	*THAN 5 ALPHANUMERIC CHARACTERS.	SFREQ	7
8		SFREQ	8
9	*****COMF		1
10	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
11	*****COMF	***COMF	3
12	INTEGER FMAX	COMF	4
13	PARAMETER (FMAX = 50)	COMF	5
14	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
15	\$ FTOT	COMF	7
16	COMMON /INITILC/ BLDG	COMF	8
17	CHARACTER * 5 BLDG	COMF	9
18	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
19	INTEGER QUALITY, FERR, FTOT	COMF	11
20	*****COMF		12
21	*****COMF		13
22	INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE	SFREQ	10
23	INTEGER IERR	SFREQ	11
24	CHARACTER * 7 PFN	SFREQ	12
25	*	SFREQ	13
26	* INITIALIZATION	SFREQ	14
27	QUIT = 0	SFREQ	15
28	FTOT = 0	SFREQ	16
29	ABORT = 0	SFREQ	17
30	100 PRINT*	SFREQ	18
31	PRINT *, 'ENTER BUILDING IDENTIFICATION (E.G. '101')'	SFREQ	19
32	PRINT *, ' (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'	SFREQ	20
33	REWIND 1	SFREQ	21
34	READ(1,*,END=100) BLDG	SFREQ	22
35		SFREQ	23
36	IF (GETLEN(BLDG) .GT. 5) THEN	SFREQ	24
37	GO TO 100	SFREQ	25
38	END IF	SFREQ	26
39	PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'F'	SFREQ	27
40	*	SFREQ	28
41	*** LOAD DATA FROM EXISTING FILE IF NECESSARY	SFREQ	29
42	200 PRINT*	SFREQ	30
43	PRINT*, 'WILL THIS BE'	SFREQ	31
44	PRINT*, ' (1) A MODIFICATION OF AN EXISTING FILE?'	SFREQ	32
45	PRINT*, ' (2) A NEW FILE?'	SFREQ	33
46	PRINT*, 'ENTER A NUMBER !!!'	SFREQ	34
47	REWIND 1	SFREQ	35
48	READ(1,*,END=200) OLDFILE	SFREQ	36
49	IF ((OLDFILE .NE. 1) .AND. (OLDFILE .NE. 2)) THEN	SFREQ	37
50	GOTO 200	SFREQ	38
51	ELSE IF (OLDFILE .EQ. 1) THEN	SFREQ	39
52	*	SFREQ	40
53	*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	SFREQ	41
54	IERR = 0	SFREQ	42
55	CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)	SFREQ	43
56	IF (IERR .EQ. 2) THEN	SFREQ	44
57	PRINT*	SFREQ	45
58	PRINT *, 'FILE ',PFN, ' NOT FOUND'	SFREQ	46
59	PRINT*, 'PROGRAM ABORTED!!!'	SFREQ	47
60	PRINT*	SFREQ	48
61	PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART '	SFREQ	49
62	+ 'PROGRAM'	SFREQ	50
63	PRINT*	SFREQ	51
64	STOP	SFREQ	52

65 *		SFREQ	53
66	ELSE	SFREQ	54
67	CALL LFREQ	SFREQ	55
68	IF (FERR .NE. 0) CALL ERROR(5)	SFREQ	56
69	END IF	SFREQ	57
70	ELSE IF (OLDFILE .EQ. 2) THEN	SFREQ	58
71 *		SFREQ	59
72 ***	CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME	SFREQ	60
73	IERR = 0	SFREQ	61
74	CALL PF ('GET',0,PFN(1:GETLEN(PFN)), 'RC', IERR)	SFREQ	62
75	IF (IERR .EQ. 0) THEN	SFREQ	63
76	PRINT*	SFREQ	64
77	PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ', BLDG	SFREQ	65
78	PRINT*	SFREQ	66
79	PRINT*, 'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',	SFREQ	67
80	+ 'OVER THE OLD FILE.'	SFREQ	68
81 250	PRINT*	SFREQ	69
82	PRINT*, 'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'	SFREQ	70
83	PRINT*, 'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'	SFREQ	71
84	REWIND 1	SFREQ	72
85	READ(1,*,END=250) ANSWER	SFREQ	73
86	IF (ANSWER .EQ. 1) THEN	SFREQ	74
87	PRINT*	SFREQ	75
88	PRINT*, 'PROGRAM HAS BEEN ABORTED, AT YOUR REQUEST'	SFREQ	76
89	PRINT*	SFREQ	77
90	STOP	SFREQ	78
91	ELSE IF (ANSWER .EQ. 2) THEN	SFREQ	79
92 9090	CONTINUE	SFREQ	80
93	ELSE	SFREQ	81
94	GOTO 250	SFREQ	82
95	END IF	SFREQ	83
96	ELSE IF (IERR .EQ. 2) THEN	SFREQ	84
97 *		SFREQ	85
98 ***	NO DATA FILE ALREADY EXISTS FOR THIS BUILDING AND DATA ENTRY	SFREQ	86
99 ***	CAN CONTINUE	SFREQ	87
100 9091	CONTINUE	SFREQ	88
101	ELSE	SFREQ	89
102 *		SFREQ	90
103 ***	PERMANENT FILE ERROR	SFREQ	91
104	PRINT*	SFREQ	92
105	PRINT*, 'PROGRAM ABORTED !!!'	SFREQ	93
106	PRINT*, ' SOME PERMANENT FILE ERROR HAS OCCURRED.'	SFREQ	94
107	PRINT*, ' DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',	SFREQ	95
108	+ 'AND TRY AGAIN'	SFREQ	96
109	STOP	SFREQ	97
110	END IF	SFREQ	98
111 *		SFREQ	99
112	PRINT*	SFREQ	100
113	PRINT*, ' BEGIN ENTERING DATA'	SFREQ	101
114 300	FTOT = FTOT + 1	SFREQ	102
115	IF (FTOT .EQ. 1) THEN	SFREQ	103
116	CALL DATAIN(1,FTOT)	SFREQ	104
117	ELSE	SFREQ	105
118	CALL DATAIN (0,FTOT)	SFREQ	106
119	END IF	SFREQ	107
120 400	PRINT*	SFREQ	108
121	PRINT*, 'DO YOU WANT TO ENTER MORE DATA? ',	SFREQ	109
122	+ '(1) YES (2) NO'	SFREQ	110
123	PRINT*, ' ENTER A NUMBER !!!'	SFREQ	111
124	REWIND 1	SFREQ	112
125	READ(1,*,END=400) ANSWER	SFREQ	113
126	IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SFREQ	114
127	GOTO 400	SFREQ	115
128	ELSE IF (ANSWER .EQ. 1) THEN	SFREQ	116

129	GOTO 300	SFREQ	117
130	ELSE IF (ANSWER .EQ. 2) THEN	SFREQ	118
131	PRINT*	SFREQ	119
132	PRINT*, 'DATA ENTRY DISCONTINUED'	SFREQ	120
133	END IF	SFREQ	121
134	END IF	SFREQ	122
135	*	SFREQ	123
136	*** MANIPULATE DATA	SFREQ	124
137	CALL MANIP (QUIT,ABORT)	SFREQ	125
138	*	SFREQ	126
139	*** TERMINATE PROGRAM, STORING DATA IF NECESSARY	SFREQ	127
140	IF (QUIT .EQ. 1) THEN	SFREQ	128
141	OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',	SFREQ	129
142	+ ACCESS='SEQUENTIAL',STATUS='NEW')	SFREQ	130
143	500 FORMAT (1PE12.6)	SFREQ	131
144	DO 600 N = 1,FTOT	SFREQ	132
145	WRITE (6,500) FREQA(N)	SFREQ	133
146	600 CONTINUE	SFREQ	134
147	ENDFILE(6)	SFREQ	135
148	CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))	SFREQ	136
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT			
149	CLOSE(6,STATUS='DELETE')	SFREQ	137
150	PRINT*	SFREQ	138
151	PRINT*, 'DATA HAS BEEN STORED AND PROGRAM TERMINATED'	SFREQ	139
152	END IF	SFREQ	140
153	IF(ABORT .EQ. 1) THEN	SFREQ	141
154	PRINT*	SFREQ	142
155	PRINT*, 'PROGRAM HAS BEEN ABORTED'	SFREQ	143
156	PRINT*, ' NO DATA HAS BEEN STORED !!!'	SFREQ	144
157	END IF	SFREQ	145
158	STOP	SFREQ	146
159	END	SFREQ	147

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	1031B		INTEGER	
AFLAG	2B	/INITILN/	REAL	
ANSWER	1032B		INTEGER	
BLDG	0B	/INITILC/	CHAR*5	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
IERR	1035B		INTEGER	
LINE	NONE	UNUSED/*S*	INTEGER	
N	1034B		INTEGER	
OLDFILE	1033B		INTEGER	
PFN	1036B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
QUIT	1030B		INTEGER	
RFLAG	3B	/INITILN/	REAL	
Y1	NONE	UNUSED/*S*	INTEGER	
Y2	NONE	UNUSED/*S*	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX INTEGER 50

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS----- -NAME-----TYPE-----ARGS-----CLASS-----

DATAIN		2	SUBROUTINE	LFREQ	0	SUBROUTINE
ERROR		1	SUBROUTINE	MANIP	2	SUBROUTINE
GETLEN	INTEGER	1	FUNCTION	PF	5	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

100	21B	30	500	570B	FORMAT	143
200	47B	42	600	INACTIVE	DO-TERM	146
250	166B	81	9090	*NO REFS*		92
300	246B	114	9091	*NO REFS*		100
400	260B	120				

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

SFREQ 14B 0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE1 FMT/SEQ
TAPE6 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH 1040B = 544
CM LABELLED COMMON LENGTH 71B = 57
CM STORAGE USED 63000B = 26112
COMPILE TIME 0.260 SECONDS

1 WARNING ERROR IN SFREQ

1	SUBROUTINE DATAIN (INSERT,LINE)	SFREQ	148
2	*****	COMF	1
3	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
4	*****	COMF	3
5	INTEGER FMAX	COMF	4
6	PARAMETER (FMAX = 50)	COMF	5
7	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
8	\$ FTOT	COMF	7
9	COMMON /INITILC/ BLDG	COMF	8
10	CHARACTER * 5 BLDG	COMF	9
11	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
12	INTEGER QUALITY, FERR, FTOT	COMF	11
13	*****	COMF	12
14	*****	COMF	13
15	INTEGER INSERT,LINE	SFREQ	150
16	IF(INSERT.EQ.1) THEN	SFREQ	151
17	200 PRINT *, ' ENTER FREQUENCY FOR LINE #',LINE	SFREQ	152
18	300 READ(1,*,END=200,ERR=200) FREQA(LINE)	SFREQ	153
19	ENDIF	SFREQ	154
20	IF(INSERT.EQ.0) THEN	SFREQ	155
21	400 PRINT *, 'ENTER NEXT FREQS, ONE PER LINE AFTER' ,	SFREQ	156
22	+ ' EACH QUESTION MARK.'	SFREQ	157
23	PRINT *, ' ENTER ZERO (0.0) TO DISCONTINUE ENTRIES'	SFREQ	158
24	PRINT *, 'START WITH LINE NUMBER = ', LINE	SFREQ	159
25	500 REWIND 1	SFREQ	160
26	READ(1,*,END=400,ERR=400) FREQA(LINE)	SFREQ	161
27	IF(FREQA(LINE).GT.0.0) THEN	SFREQ	162
28	*****	SFREQ	163
29	* CHECK IF ARRAY SIZE EXCEEDED	SFREQ	164
30	*****	SFREQ	165
31	IF(LINE.GT.FMAX) THEN	SFREQ	166
32	PRINT *, 'MAXIMUM NUMBER OF DATA LINES CANNOT '	SFREQ	167
33	PRINT *, 'EXCEED ', FMAX, '. INSERTION NOT POSSIBLE.'	SFREQ	168
34	RETURN	SFREQ	169
35	ENDIF	SFREQ	170
36	*****	SFREQ	171
37	LINE = LINE + 1	SFREQ	172
38	GOTO 500	SFREQ	173
39	ELSE	SFREQ	174
40	LINE =LINE -1	SFREQ	175
41	ENDIF	SFREQ	176
42	ENDIF	SFREQ	177
43	RETURN	SFREQ	178
44	END	SFREQ	179

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
INSERT	1	DUMMY-ARG	INTEGER	
LINE	2	DUMMY-ARG	INTEGER	
QUALITY	1B	/INITILN/	INTEGER	
RFLAG	3B	/INITILN/	REAL	

--SYMBOLIC CONSTANTS--(LO=A)
-NAME---TYPE-----VALUE

FMAX INTEGER 50

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

200 12E 17
300 *NO REFS* 18
400 25B 21
500 33B 25

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

DATAIN 5B 2

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE1 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH 200B = 128
CM LABELLED COMMON LENGTH 71B = 57
CM STORAGE USED 61000B = 25088
COMPILE TIME 0.073 SECONDS

1	SUBROUTINE MANIP (QUIT,ABORT)	SFREQ	180
2	*****	COMF	1
3	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
4	*****	COMF	3
5	INTEGER FMAX	COMF	4
6	PARAMETER (FMAX = 50)	COMF	5
7	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
8	5 FTOT	COMF	7
9	COMMON /INITILC/ BLDG	COMF	8
10	CHARACTER * 5 BLDG	COMF	9
11	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
12	INTEGER QUALITY, FERR, FTOT	COMF	11
13	*****	COMF	12
14	*****	COMF	13
15	INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT	SFREQ	182
16	INTEGER TEMP,V,X,Y,COMMAND	SFREQ	183
17	CHARACTER * 3 DIR, FROM, TO	SFREQ	184
18	*	SFREQ	185
19	10 FLAG1 = 0	SFREQ	186
20	PRINT*	SFREQ	187
21	PRINT*, 'CHOOSE'	SFREQ	188
22	PRINT*, ' (1) DISPLAY LINE OF DATA (4) DISPLAY ALL LINES'	SFREQ	189
23	PRINT*, ' (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'	SFREQ	190
24	PRINT*, ' (3) DELETE LINE (6) STORE DATA AND EXIT ',	SFREQ	191
25	+ 'PROGRAM'	SFREQ	192
26	PRINT*, ' (7) EXIT PROGRAM WITHOUT ',	SFREQ	193
27	+ 'STORING DATA'	SFREQ	194
28	PRINT*, 'ENTER A NUMBER !!!'	SFREQ	195
29	PRINT*	SFREQ	196
30	REWIND 1	SFREQ	197
31	READ(1,*,END=10) COMMAND	SFREQ	198
32	*	SFREQ	199
33	*-----	SFREQ	200
34	*** DISPLAY LINE ***	SFREQ	201
35	*-----	SFREQ	202
36	IF (COMMAND .EQ. 1) THEN	SFREQ	203
37	*	SFREQ	204
38	*** INDICATE EMPTY DATA FILE	SFREQ	205
39	IF (FTOT .EQ. 0) THEN	SFREQ	206
40	PRINT*	SFREQ	207
41	PRINT*, 'DATA FILE IS EMPTY !!!'	SFREQ	208
42	*	SFREQ	209
43	*** ENTER NUMBER OF LINE TO BE DISPLAYED	SFREQ	210
44	ELSE	SFREQ	211
45	100 PRINT*	SFREQ	212
46	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'	SFREQ	213
47	PRINT*, ' (ENTER "0" TO ESCAPE DISPLAY MODE)'	SFREQ	214
48	REWIND 1	SFREQ	215
49	READ(1,*,END=100) N	SFREQ	216
50	*	SFREQ	217
51	*** CHECK VALIDITY OF LINE NUMBER	SFREQ	218
52	IF ((N .GT. FTOT) .OR. (N .LT. 0)) THEN	SFREQ	219
53	PRINT*	SFREQ	220
54	PRINT*, 'INCORRECT NUMBER !!!!! TRY AGAIN !!!'	SFREQ	221
55	PRINT*, ' -OR- ENTER "0" TO ESCAPE FROM ',	SFREQ	222
56	+ '"DISPLAY" MODE'	SFREQ	223
57	GOTO 100	SFREQ	224
58	*	SFREQ	225
59	*** ABORT 'DISPLAY' MODE	SFREQ	226
60	ELSE IF (N .EQ. 0) THEN	SFREQ	227
61	PRINT*	SFREQ	228
62	PRINT*, ' "DISPLAY" MODE ABORTED !!!'	SFREQ	229
63	*	SFREQ	230
64	*** DISPLAY LINE OF DATA	SFREQ	231

65	ELSE IF ((N .GT. 0) .AND. (N .LE. FTOT)) THEN	SFREQ	232
66	PRINT*	SFREQ	233
67	CALL DISPLAY(N, COMMAND)	SFREQ	234
68 *		SFREQ	235
69	END IF	SFREQ	236
70	END IF	SFREQ	237
71	END IF	SFREQ	238
72 *		SFREQ	239
73 *	-----	SFREQ	240
74 *** INSERT LINE ***		SFREQ	241
75 *	-----	SFREQ	242
76	IF (COMMAND .EQ. 2) THEN	SFREQ	243
77 *		SFREQ	244
78 *** INDICATE EMPTY DATA FILE		SFREQ	245
79	IF (FTOT .EQ. 0) THEN	SFREQ	246
80	PRINT*	SFREQ	247
81	PRINT*, 'DATA FILE IS EMPTY !!!'	SFREQ	248
82 *		SFREQ	249
83 *** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE		SFREQ	250
84	ELSE	SFREQ	251
85 200	PRINT*	SFREQ	252
86	PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',	SFREQ	253
87	+	SFREQ	254
	'TO BE INSERTED'		
88	PRINT*, ' (ENTER "0" TO ESCAPE "INSERTION" MODE) '	SFREQ	255
89	REWIND 1	SFREQ	256
90	READ(1,*,END=200) N	SFREQ	257
91 *		SFREQ	258
92 *** CHECK FOR VALID LINE NUMBER		SFREQ	259
93	IF ((N .LT. 0) .OR. (N .GT. FTOT)) THEN	SFREQ	260
94	PRINT*	SFREQ	261
95	PRINT*, 'INCORRECT LINE NUMBER !!!'	SFREQ	262
96	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE',	SFREQ	263
97	+	SFREQ	264
	'"INSERTION" MODE'		
98	GOTO 200	SFREQ	265
99 *		SFREQ	266
100 *** ABORT INSERTION MODE		SFREQ	267
101	ELSE IF (N .EQ. 0) THEN	SFREQ	268
102	PRINT*	SFREQ	269
103	PRINT*, ' "INSERTION" MODE ABORTED'	SFREQ	270
104 *		SFREQ	271
105 *** MAKE ROOM FOR NEW LINE OF DATA		SFREQ	272
106	ELSE IF ((N .GT. 0) .AND. (N .LE. FTOT)) THEN	SFREQ	273
107	*****	SFREQ	274
108 *	CHECK IF ARRAY SIZE EXCEEDED	SFREQ	275
109	*****	SFREQ	276
110	IF(FTOT.EQ.FMAX) THEN	SFREQ	277
111	PRINT *, 'MAXIMUM NUMBER OF DATA LINES CANNOT '	SFREQ	278
112	PRINT *, 'EXCEED ', FMAX, ' . INSERTION NOT POSSIBLE.'	SFREQ	279
113	GO TO 10	SFREQ	280
114	ENDIF	SFREQ	281
115	*****	SFREQ	282
116	DO 230 X = FTOT,N,-1	SFREQ	283
117	FREQA(X+1) = FREQA(X)	SFREQ	284
118 210	CONTINUE	SFREQ	285
119 230	CONTINUE	SFREQ	286
120 *		SFREQ	287
121 *** ENTER DATA FOR NEW LINE		SFREQ	288
122	FTOT = FTOT + 1	SFREQ	289
123	CALL DATAIN (1,N)	SFREQ	290
124 *		SFREQ	291
125	PRINT*	SFREQ	292
126	PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N	SFREQ	293
127	CALL DISPLAY(N, COMMAND)	SFREQ	294
128	END IF	SFREQ	295

129	END IF	SFREQ	296
130	END IF	SFREQ	297
131	*	SFREQ	298
132	-----	SFREQ	299
133	*** DELETE LINE ***	SFREQ	300
134	-----	SFREQ	301
135	IF (COMMAND .EQ. 3) THEN	SFREQ	302
136	*	SFREQ	303
137	*** INDICATE EMPTY DATA FILE	SFREQ	304
138	IF (FTOT .EQ. 0) THEN	SFREQ	305
139	PRINT*	SFREQ	306
140	PRINT*, 'DATA FILE IS EMPTY !!!'	SFREQ	307
141	*	SFREQ	308
142	*** READ NUMBER OF LINE TO BE DELETED	SFREQ	309
143	ELSE	SFREQ	310
144	300 PRINT*	SFREQ	311
145	PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'	SFREQ	312
146	PRINT*, ' (ENTER "0" TO ESCAPE DELETION MODE)'	SFREQ	313
147	REWIND 1	SFREQ	314
148	READ(1,*,END=300) N	SFREQ	315
149	*	SFREQ	316
150	*** CHECK VALIDITY OF LINE NUMBER	SFREQ	317
151	IF ((N .GT. FTOT) .OR. (N .LT. 0)) THEN	SFREQ	318
152	PRINT*	SFREQ	319
153	PRINT*, ' INCORRECT NUMBER !!!'	SFREQ	320
154	PRINT*, ' TRY AGAIN !!! -OR- ENTER "0" TO ESCAPE FROM'	SFREQ	321
155	+ "DELETE" MODE'	SFREQ	322
156	GOTO 300	SFREQ	323
157	*	SFREQ	324
158	*** ABORT 'DELETE' MODE	SFREQ	325
159	ELSE IF (N .EQ. 0) THEN	SFREQ	326
160	PRINT*, ' "DELETE" MODE ABORTED'	SFREQ	327
161	*	SFREQ	328
162	*** DOUBLE CHECK CHOICE OF LINE TO BE DELETED	SFREQ	329
163	ELSE IF ((N .GT. 0) .AND. (N .LE. FTOT)) THEN	SFREQ	330
164	PRINT*	SFREQ	331
165	PRINT*, 'DOUBLE CHECK !!!'	SFREQ	332
166	PRINT*, ' DO YOU WANT TO DELETE THE FOLLOWING LINE?:'	SFREQ	333
167	CALL DISPLAY(N, COMMAND)	SFREQ	334
168	305 PRINT*, ' ENTER (1) YES OR (2) NO'	SFREQ	335
169	REWIND 1	SFREQ	336
170	READ(1,*,END=305) ANSWER	SFREQ	337
171	*	SFREQ	338
172	*** DELETE LINE	SFREQ	339
173	IF (ANSWER .EQ. 1) THEN	SFREQ	340
174	DO 330 X = N, FTOT - 1	SFREQ	341
175	FREQA(X) = FREQA(X+1)	SFREQ	342
176	330 CONTINUE	SFREQ	343
177	FTOT = FTOT - 1	SFREQ	344
178	PRINT*	SFREQ	345
179	PRINT*, 'LINE # ',N,' DELETED'	SFREQ	346
180	END IF	SFREQ	347
181	*	SFREQ	348
182	END IF	SFREQ	349
183	END IF	SFREQ	350
184	END IF	SFREQ	351
185	*	SFREQ	352
186	-----	SFREQ	353
187	*** DISPLAY ALL DATA ***	SFREQ	354
188	-----	SFREQ	355
189	IF (COMMAND .EQ. 4) THEN	SFREQ	356
190	*	SFREQ	357
191	*** INDICATE EMPTY DATA FILE	SFREQ	358
192	IF (FTOT .EQ. 0) THEN	SFREQ	359

193	PRINT*	SFREQ	360
194	PRINT*, 'DATA FILE IS EMPTY !!!'	SFREQ	361
195 *		SFREQ	362
196 *** DISPLAY DATA		SFREQ	363
197 ELSE		SFREQ	364
198 PRINT*		SFREQ	365
199 CALL DISPLAY(N, COMMAND)		SFREQ	366
200 *		SFREQ	367
201 END IF		SFREQ	368
202 END IF		SFREQ	369
203 *		SFREQ	370
204 *-----		SFREQ	371
205 *** ADD DATA ***		SFREQ	372
206 *-----		SFREQ	373
207 IF (COMMAND .EQ. 5) THEN		SFREQ	374
208 *		SFREQ	375
209 *** ENTER DATA		SFREQ	376
210 500 FTOT = FTOT + 1		SFREQ	377
211 *****		SFREQ	378
212 * CHECK IF ARRAY SIZE EXCEEDED		SFREQ	379
213 *****		SFREQ	380
214 IF(FTOT.EQ.FMAX) THEN		SFREQ	381
215 PRINT *, 'MAXIMUM NUMBER OF DATA LINES CANNOT '		SFREQ	382
216 PRINT *, 'EXCEED ', FMAX, '. INSERTION NOT POSSIBLE.'		SFREQ	383
217 GO TO 10		SFREQ	384
218 ENDIF		SFREQ	385
219 *****		SFREQ	386
220 CALL DATAIN (0,FTOT)		SFREQ	387
221 510 PRINT*		SFREQ	388
222 PRINT*, 'DO YOU WANT TO ENTER MORE DATA? (1) YES (2) NO'		SFREQ	389
223 PRINT*, ' ENTER A NUMBER !!!'		SFREQ	390
224 REWIND 1		SFREQ	391
225 READ(1,*,END=510) ANSWER		SFREQ	392
226 *		SFREQ	393
227 *** CHECK VALIDITY OF NUMBER		SFREQ	394
228 IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN		SFREQ	395
229 GOTO 510		SFREQ	396
230 *		SFREQ	397
231 *** ENTER MORE DATA		SFREQ	398
232 ELSE IF (ANSWER .EQ. 1) THEN		SFREQ	399
233 GOTO 500		SFREQ	400
234 *		SFREQ	401
235 *** DISCONTINUE DATA ENTRY		SFREQ	402
236 ELSE IF (ANSWER .EQ. 2) THEN		SFREQ	403
237 PRINT*		SFREQ	404
238 PRINT*, 'DATA ENTRY DISCONTINUED'		SFREQ	405
239 *		SFREQ	406
240 END IF		SFREQ	407
241 END IF		SFREQ	408
242 *		SFREQ	409
243 *-----		SFREQ	410
244 *** STORE DATA AND PROGRAM ***		SFREQ	411
245 *-----		SFREQ	412
246 IF (COMMAND .EQ. 6) THEN		SFREQ	413
247 600 PRINT*		SFREQ	414
248 PRINT*, 'DOUBLE CHECK !!!'		SFREQ	415
249 PRINT*, ' DO YOU WANT TO STORE THIS DATA AND END PROG'		SFREQ	416
250 PRINT*, ' NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE '		SFREQ	417
251 PRINT*, ' OF THE SAME NAME !!!'		SFREQ	418
252 PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'		SFREQ	419
253 REWIND 1		SFREQ	420
254 READ(1,*,END=600) ANSWER		SFREQ	421
255 *		SFREQ	422
256 *** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM		SFREQ	423

257	IF (ANSWER .EQ. 1) THEN	SFREQ	424
258	QUIT = 1	SFREQ	425
259	RETURN	SFREQ	426
260 *		SFREQ	427
261 ***	ABORT 'STORING' MODE	SFREQ	428
262	ELSE IF (ANSWER .EQ. 2) THEN	SFREQ	429
263	PRINT*	SFREQ	430
264	PRINT*, ' "STORING" MODE DISCONTINUED'	SFREQ	431
265 *		SFREQ	432
266 ***	CHECK VALIDITY OF ANSWER	SFREQ	433
267	ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SFREQ	434
268	GOTO 600	SFREQ	435
269 *		SFREQ	436
270	END IF	SFREQ	437
271	END IF	SFREQ	438
272 *		SFREQ	439
273 *	-----	SFREQ	440
274 ***	END PROGRAM WITHOUT STORING DATA ***	SFREQ	441
275 *	-----	SFREQ	442
276	IF (COMMAND .EQ. 7) THEN	SFREQ	443
277 700	PRINT*	SFREQ	444
278	PRINT*, 'DOUBLE CHECK !!!'	SFREQ	445
279	PRINT*, ' DO YOU WANT TO END THIS PROGRAM ',	SFREQ	446
280 +	'WITHOUT STORING DATA?'	SFREQ	447
281	PRINT*, ' ENTER A NUMBER: (1) YES (2) NO'	SFREQ	448
282	REWIND 1	SFREQ	449
283	READ(1,*,END=700) ANSWER	SFREQ	450
284 *		SFREQ	451
285 ***	SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM	SFREQ	452
286	IF (ANSWER .EQ. 1) THEN	SFREQ	453
287	ABORT = 1	SFREQ	454
288	RETURN	SFREQ	455
289 *		SFREQ	456
290 ***	ABORT 'STORING' MODE	SFREQ	457
291	ELSE IF (ANSWER .EQ. 2) THEN	SFREQ	458
292	PRINT*	SFREQ	459
293	PRINT*, ' "ABORTION" MODE DISCONTINUED'	SFREQ	460
294 *		SFREQ	461
295 ***	CHECK VALIDITY OF ANSWER	SFREQ	462
296	ELSE IF ((ANSWER .NE. 1) .AND. (ANSWER .NE. 2)) THEN	SFREQ	463
297	GOTO 700	SFREQ	464
298 *		SFREQ	465
299	END IF	SFREQ	466
300	END IF	SFREQ	467
301 *		SFREQ	468
302 *	-----	SFREQ	469
303 ***	LOOP TO BEGINNING OF 'MANIP' SUBROUTINE	SFREQ	470
304 *	-----	SFREQ	471
305	GOTO 10	SFREQ	472
306 *		SFREQ	473
307	END	SFREQ	474

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ABORT	2	DUMMY-ARG	INTEGER
AFLAG	2B	/INITILN/	REAL
ANSWER	1364B		INTEGER
BLDG	0B	/INITILC/	CHAR*5
COMMAND	1370B		INTEGER
DIR	NONE	UNUSED/*S*	CHAR*3
DOK	NONE	UNUSED/*S*	INTEGER
FERR	66B	/INITILN/	INTEGER

FLAG1	1365B		INTEGER
FREQ	0B	/INITILN/	REAL
FREQA	4B	/INITILN/	REAL 50
FROM	NONE	UNUSED/*S*	CHAR*3
FTOT	67B	/INITILN/	INTEGER
INSERT	NONE	UNUSED/*S*	INTEGER
LOK	NONE	UNUSED/*S*	INTEGER
N	1366B		INTEGER
NOK	NONE	UNUSED/*S*	INTEGER
OK	NONE	UNUSED/*S*	INTEGER
OK1	NONE	UNUSED/*S*	INTEGER
OK2	NONE	UNUSED/*S*	INTEGER
QUALITY	1B	/INITILN/	INTEGER
QUIT	1	DUMMY-ARG	INTEGER
RFLAG	3B	/INITILN/	REAL
TEMP	NONE	UNUSED/*S*	INTEGER
TO	NONE	UNUSED/*S*	CHAR*3
V	NONE	UNUSED/*S*	INTEGER
X	1367B		INTEGER
Y	NONE	UNUSED/*S*	INTEGER

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

DATAIN	2	SUBROUTINE
DISPLAY	2	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF -LABEL-ADDRESS-----PROPERTIES----DEF

10	7B	19	305	325B	168
100	50B	45	330	INACTIVE DO-TERM	176
200	133B	85	500	413B	210
210	*NO REFS*	118	510	427B	221
230	INACTIVE DO-TERM	119	600	470B	247
300	256B	144	700	542B	277

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

MANIP	5B	2
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE1	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	1375B = 765
CM LABELLED COMMON LENGTH	71B = 57
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.406 SECONDS

1	SUBROUTINE DISPLAY (LINE, COMMAND)	SFREQ	475
2	*****	COMF	1
3	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
4	*****	COMF	3
5	INTEGER FMAX	COMF	4
6	PARAMETER (FMAX = 50)	COMF	5
7	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
8	\$ FTOT	COMF	7
9	COMMON /INITILC/ BLDG	COMF	8
10	CHARACTER * 5 BLDG	COMF	9
11	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
12	INTEGER QUALITY, FERR, FTOT	COMF	11
13	*****	COMF	12
14	*****	COMF	13
15	INTEGER LINE, COMMAND, N	SFREQ	477
16	1000 FORMAT (1X, 'LINE # FREQUENCY (HZ)')	SFREQ	478
17	2000 FORMAT (4X, I3, 8X, 1PE15.5)	SFREQ	479
18	PRINT 1000	SFREQ	480
19	IF (COMMAND .EQ. 4) THEN	SFREQ	481
20	DO 10 N = 1, FTOT	SFREQ	482
21	PRINT 2000, N, FREQA(N)	SFREQ	483
22	10 CONTINUE	SFREQ	484
23	ELSE	SFREQ	485
24	PRINT 2000, LINE, FREQA(LINE)	SFREQ	486
25	END IF	SFREQ	487
26	RETURN	SFREQ	488
27	END	SFREQ	489

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL
BLDG	0B	/INITILC/	CHAR*5
COMMAND	2	DUMMY-ARG	INTEGER
FERR	66B	/INITILN/	INTEGER
FREQ	0B	/INITILN/	REAL
FREQA	4B	/INITILN/	REAL 50
FTOT	67B	/INITILN/	INTEGER
LINE	1	DUMMY-ARG	INTEGER
N	100B		INTEGER
QUALITY	1B	/INITILN/	INTEGER
RFLAG	3B	/INITILN/	REAL

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
------	---------	----

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	22
1000	51B	FORMAT	16
2000	56B	FORMAT	17

FTN 5.1+552 83/12/20 11.52.59 PAGE 15
SUBROUTINE DISPLAY 74/175 OPT=0

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

DISPLAY	5B	2
---------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	104B = 68
CM LABELLED COMMON LENGTH	71B = 57
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.045 SECONDS

1	INTEGER FUNCTION VAL(String)	SFREQ	490
2	C** RETURNS THE INTEGER VALUE OF A STRING.	SFREQ	491
3	INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN	SFREQ	492
4	CHARACTER * (*) STRING	SFREQ	493
5	L = GETLEN(String)	SFREQ	494
6	NUMBER = 0	SFREQ	495
7	DO 10 X = L,1,-1	SFREQ	496
8	EXP = L - X	SFREQ	497
9	DIGIT = ICHAR(String(X:X)) - 16	SFREQ	498
10	NUMBER = NUMBER + DIGIT*10**EXP	SFREQ	499
11	10 CONTINUE	SFREQ	500
12	VAL = NUMBER	SFREQ	501
13	RETURN	SFREQ	502
14	END	SFREQ	503

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

DIGIT	76B		INTEGER
EXP	75B		INTEGER
L	74B		INTEGER
NUMBER	72B		INTEGER
STRING	1	DUMMY-ARG	CHAR*(*)
VAL	71B		INTEGER
X	73B		INTEGER

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	11
----	----------	---------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

VAL	6B	1
-----	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.039 SECONDS

1	INTEGER FUNCTION GETLEN (STRING)	SFREQ	504
2	C	SFREQ	505
3	C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING	SFREQ	506
4	C	SFREQ	507
5	C	SFREQ	508
6	C ARGUMENT DEFINITIONS --	SFREQ	509
7	C READ ARGUMENTS	SFREQ	510
8	C STRING - STRING WHOSE LENGTH IS TO BE DETERMINED	SFREQ	511
9	C	SFREQ	512
10	CHARACTER * (*) STRING	SFREQ	513
11	C	SFREQ	514
12	C FUNCTION PARAMETERS	SFREQ	515
13	CHARACTER * 1 BLANK	SFREQ	516
14	PARAMETER (BLANK = ' ')	SFREQ	517
15	C	SFREQ	518
16	C LOCAL VARIABLES	SFREQ	519
17	INTEGER NEXT	SFREQ	520
18	C	SFREQ	521
19	C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK	SFREQ	522
20	DO 10 NEXT = LEN(STRING),1,-1	SFREQ	523
21	IF (STRING(NEXT : NEXT) .NE. BLANK) THEN	SFREQ	524
22	GETLEN = NEXT	SFREQ	525
23	RETURN	SFREQ	526
24	END IF	SFREQ	527
25	10 CONTINUE	SFREQ	528
26	C	SFREQ	529
27	C ALL CHARACTERS ARE BLANKS	SFREQ	530
28	GETLEN = 0	SFREQ	531
29	C	SFREQ	532
30	RETURN	SFREQ	533
31	END	SFREQ	534

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

GETLEN	63B	INTEGER
NEXT	64B	INTEGER
STRING	1 DUMMY-ARG	CHAR*(*)

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

BLANK	CHAR*1	' '
-------	--------	-----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

LEN	INTEGER	1	INTRINSIC
-----	---------	---	-----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	25
----	----------	---------	----

FTN 5.1+552 83/12/20. 11.52.59 PAGE 18
 FUNCTION GETLEN 74/175 OPT=0

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

GETLEN	6B	1
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	70B = 56
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.039 SECONDS

```

1      SUBROUTINE LFREQ                                LFREQ      1
2      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LFREQ      2
3      *!!!!                                           !!!LFREQ      3
4      *!!!!   LOAD THE CONTENTS OF THE FILE 'BXXXXXF' INTO ARRAYS FREQA.   LFREQ      4
5      *!!!!                                           !!!LFREQ      5
6      *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! LFREQ      6
7      *****LFREQ      7
8      *****COMF      1
9      *** COMMON FOR INITIAL PARAMETERS                ***COMF      2
10     *****COMF      3
11     INTEGER FMAX                                     COMF      4
12     PARAMETER (FMAX = 50)                            COMF      5
13     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
14     $          FTOT                                   COMF      7
15     COMMON /INITILC/ BLDG                             COMF      8
16     CHARACTER * 5 BLDG                                COMF      9
17     REAL FREQ, AFLAG, RFLAG, FREQA                    COMF     10
18     INTEGER QUALITY, FERR, FTOT                       COMF     11
19     *****COMF     12
20     *****COMF     13
21     *****LFREQ     9
22     *   DECLARATION OF VARIABLES                     LFREQ     10
23     *****LFREQ     11
24     INTEGER GETLEN, R, C                              LFREQ     12
25     CHARACTER * 7 NAME, PFN                           LFREQ     13
26     *****LFREQ     14
27     *                                                  LFREQ     15
28     *****LFREQ     16
29     NAME = 'B'//BLDG(1:GETLEN(BLDG))//'F'            LFREQ     17
30     PFN = NAME (1:GETLEN(NAME))                      LFREQ     18
31     FERR = 0                                           LFREQ     19
32     CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',FERR)    LFREQ     20
33     IF ( FERR .EQ. 0 ) THEN                          LFREQ     21
34         OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',    LFREQ     22
35         $      STATUS='OLD', ACCESS='SEQUENTIAL')    LFREQ     23
36         FTOT = 0                                       LFREQ     24
37         DO 10 R = 1,FMAX                              LFREQ     25
38             READ (3,1000,END=20) FREQA(R)            LFREQ     26
39 1000     FORMAT(E12.7)                                LFREQ     27
40             FTOT = FTOT + 1                          LFREQ     28
41 10     CONTINUE                                       LFREQ     29
42 20     CONTINUE                                       LFREQ     30
43         CLOSE(3,STATUS='DELETE')                    LFREQ     31
44     ELSE IF ( FERR .EQ. 2 ) THEN                     LFREQ     32
45         CALL WARNING (11)                            LFREQ     33
46     ELSE                                             LFREQ     34
47         CALL WARNING (12)                            LFREQ     35
48     END IF                                           LFREQ     36
49     RETURN                                           LFREQ     37
50     END                                              LFREQ     38

```

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	NONE	UNUSED/*S*	INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
NAME	210B		CHAR*7	
PFN	211B		CHAR*7	

FTN 5.1+552 83/12/20. 11.52.59 PAGE 20
 SUBROUTINE LFREQ 74/175 OPT=0

QUALITY	1B	/INITILN/	INTEGER
R	207B		INTEGER
RFLAG	3B	/INITILN/	REAL

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	41
20	63B		42
1000	120B	FORMAT	39

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

LFREQ	5B	0
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE3	AUX/FMT/SEQ
-------	-------------

--STATISTICS--

PROGRAM-UNIT LENGTH	215B = 141
CM LABELLED COMMON LENGTH	71B = 57
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.080 SECONDS

1	SUBROUTINE ERROR(IERR)	ERROR	1
2	CHARACTER*45 MESSAGE(20)	ERROR	2
3	DATA MESSAGE(1) //'MATERIALS DATA BASE IS EMPTY	// ERROR	3
4	DATA MESSAGE(2) //'FREQUENCY IS OUT OF RANGE	// ERROR	4
5	DATA MESSAGE(3) //'THIS MATERIAL IS NOT IN DATA BASE	// ERROR	5
6	DATA MESSAGE(4) //'DENOMINATOR IS ZERO	// ERROR	6
7	DATA MESSAGE(5) //'FILE HANDLING ERROR	// ERROR	7
8	DATA MESSAGE(6) //'ERROR CODE IS OUT OF RANGE	// ERROR	8
9	DATA MESSAGE(7) //'ERROR CODE IS OUT OF RANGE	// ERROR	9
10	DATA MESSAGE(8) //'ERROR CODE IS OUT OF RANGE	// ERROR	10
11	DATA MESSAGE(9) //'ERROR CODE IS OUT OF RANGE	// ERROR	11
12	DATA MESSAGE(10) //'ERROR CODE IS OUT OF RANGE	// ERROR	12
13	DATA MESSAGE(11) //'ERROR CODE IS OUT OF RANGE	// ERROR	13
14	DATA MESSAGE(12) //'ERROR CODE IS OUT OF RANGE	// ERROR	14
15	DATA MESSAGE(13) //'ERROR CODE IS OUT OF RANGE	// ERROR	15
16	DATA MESSAGE(14) //'ERROR CODE IS OUT OF RANGE	// ERROR	16
17	DATA MESSAGE(15) //'ERROR CODE IS OUT OF RANGE	// ERROR	17
18	DATA MESSAGE(16) //'ERROR CODE IS OUT OF RANGE	// ERROR	18
19	DATA MESSAGE(17) //'ERROR CODE IS OUT OF RANGE	// ERROR	19
20	DATA MESSAGE(18) //'ERROR CODE IS OUT OF RANGE	// ERROR	20
21	DATA MESSAGE(19) //'ERROR CODE IS OUT OF RANGE	// ERROR	21
22	DATA MESSAGE(20) //'ERROR CODE IS OUT OF RANGE	// ERROR	22
23	IERRM=5	ERROR	23
24	IF(IERR.GT.IERRM) IERR=20	ERROR	24
25	WRITE(6,10) IERR,MESSAGE(IERR)	ERROR	25
26	10 FORMAT(' ***ERROR NUMBER = ',I5,' *** ',A45)	ERROR	26
27	CALL PMDSTOP	ERROR	27
28	STOP 'ERROR'	ERROR	28
29	END	ERROR	29

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

IERR	1	DUMMY-ARG	INTEGER	
IERRM	210B		INTEGER	
MESSAGE	56B		CHAR*45	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PMDSTOP	0	SUBROUTINE
---------	---	------------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	36B	FORMAT	26
----	-----	--------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

ERROR	5B	1
-------	----	---

FTN 5.1+552 83/12/20 11.52.59 PAGE 22
SUBROUTINE ERROR 74/175 OPT=0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6	FMT/SEQ
-------	---------

--STATISTICS--

PROGRAM-UNIT LENGTH	2138 = 139
CM STORAGE USED	61000B = 25088
COMPILE TIME	0 056 SECONDS

1	SUBROUTINE WARNING(ERR)	WARNING	1
2	INTEGER ERR, ERRM	WARNING	2
3	CHARACTER*45 MESSAGE(20)	WARNING	3
4	DATA MESSAGE(1)/'HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	4
5	DATA MESSAGE(2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE	/ WARNING	5
6	DATA MESSAGE(3)/'MATTER" FILE DOES NOT EXIST FOR THIS BLDG	/ WARNING	6
7	DATA MESSAGE(4)/'FILE HANDLING PROBLEM ON "MATTER FILE	/ WARNING	7
8	DATA MESSAGE(5)/'TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	8
9	DATA MESSAGE(6)/'FILE HANDLING PROBLEM ON "TYPE" FILE	/ WARNING	9
10	DATA MESSAGE(7)/'WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	10
11	DATA MESSAGE(8)/'FILE HANDLING PROBLEM ON "WALL" FILE	/ WARNING	11
12	DATA MESSAGE(9)/'HEIGHT AND WIDTH OF ROOM MISSING	/ WARNING	12
13	DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING	/ WARNING	13
14	DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG	/ WARNING	14
15	DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE	/ WARNING	15
16	DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE	/ WARNING	16
17	DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE	/ WARNING	17
18	DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE	/ WARNING	18
19	DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE	/ WARNING	19
20	DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE	/ WARNING	20
21	DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE	/ WARNING	21
22	DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE	/ WARNING	22
23	DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE	/ WARNING	23
24	ERRM=12	WARNING	24
25	IERR = ERR	WARNING	25
26	IF(ERR.GT.ERRM) IERR=20	WARNING	26
27	WRITE(6,20)	WARNING	27
28	WRITE(6,10) ERR,MESSAGE(IERR)	WARNING	28
29	WRITE(6,20)	WARNING	29
30 10	FORMAT(' ***WARNING NUMBER = ',I5,' *** ',A45)	WARNING	30
31 20	FORMAT(' ')	WARNING	31
32	RETURN	WARNING	32
33	END	WARNING	33

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

ERR	1	DUMMY-ARG	INTEGER	
ERRM	60B		INTEGER	
IERR	213B		INTEGER	
MESSAGE	61B		CHAR*45	20

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES-----DEF

10	34B	FORMAT	30
20	42B	FORMAT	31

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

WARNING	5B	1
---------	----	---

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	216B = 142
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.064 SECONDS

Appendix 9.7 Listing of Computer Program MASTER

```

1      PROGRAM MASTER (INPUT,OUTPUT, TAPE1 = INPUT,TAPE6 = OUTPUT)      MASTER      1
2      *****MASTER      2
3      *****COMF      1
4      *** COMMON FOR INITIAL PARAMETERS      ***COMF      2
5      *****COMF      3
6      INTEGER FMAX      COMF      4
7      PARAMETER (FMAX = 50)      COMF      5
8      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,      COMF      6
9      5 FTOT      COMF      7
10     COMMON /INITILC/ BLDG      COMF      8
11     CHARACTER * 5 BLDG      COMF      9
12     REAL FREQ, AFLAG, RFLAG, FREQA      COMF     10
13     INTEGER QUALITY, FERR, FTOT      COMF     11
14     *****COMF     12
15     *****COMF     13
16     *****COMW      1
17     *** COMMON FOR DATABASE OF WALL PARAMETERS      ***COMW      2
18     *****COMW      3
19     INTEGER WMAX      COMW      4
20     PARAMETER (WMAX = 75)      COMW      5
21     COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR      COMW      6
22     COMMON /WALLC/ WALL(WMAX,4)      COMW      7
23     INTEGER WTOT,WERR      COMW      8
24     REAL WDIM      COMW      9
25     CHARACTER *3 WALL      COMW     10
26 * =====      COMW     11
27 * ** DESCRIPTION OF ARRAYS      COMW     12
28 * =====      COMW     13
29 * WALL IDENTIFICATION      COMW     14
30 * -----      COMW     15
31 * DIRECTION      FROM      TO      COMW     16
32 *      ROOM      ROOM      COMW     17
33 * -----      COMW     18
34 * WALL(X,1)      WALL(X,2)      WALL(X,3)      COMW     19
35 * A3      A3      A3      COMW     20
36 * =====      COMW     21
37 * WALL PARAMETERS      COMW     22
38 * -----      COMW     23
39 * MATERIAL      HEIGHT      WIDTH      LAYER THICKNESS      COMW     24
40 * -----      COMW     25
41 * WALL(X,4)      WDIM(X,1)      WDIM(X,2)      WDIM(X,3)      COMW     26
42 * A3      F8.2      F8.2      F8.2      COMW     27
43 *****COMW     28
44 *****COMW     29
45 *****COMT      1
46 *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS      ***COMT      2
47 *****COMT      3
48 INTEGER TMAX      COMT      4
49 PARAMETER (TMAX=35)      COMT      5
50 COMMON /TYPEEN/ TDIM(TMAX,4), TTOT, TDB2(TMAX,2), TDBTOT, TERR      COMT      6
51 COMMON /TYPEEC/ TYPE(TMAX,3), TDB1(TMAX)      COMT      7
52 INTEGER TTOT,TDBTOT,TERR      COMT      8
53 REAL TDIM,TDB2      COMT      9
54 CHARACTER * 3 TYPE,TDB1      COMT     10
55 *=====      COMT     11
56 * ** DESCRIPTION OF ARRAYS      COMT     12
57 *=====      COMT     13
58 * ID      MATERIAL      FRAME MATERIAL      COMT     14
59 *-----      COMT     15
60 *TYPE(X,1)      TYPE(X,2)      TYPE(X,3)      COMT     16
61 * A3      A3      A3      COMT     17
62 *=====      COMT     18
63 * HEIGHT      WIDTH      LAYER      DISTANCE      COMT     19
64 *      THICKNESS      ABOVE FLOOR      COMT     20

```

```

65 *----- COMT 21
66 * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4) COMT 22
67 * F8.2 F8.2 F8.2 F8.2 COMT 23
68 *===== COMT 24
69 * ID ATTENUATION AREA COMT 25
70 *----- COMT 26
71 * TDB1(X) TDB2(X,1) TDB2(X,2) COMT 27
72 * A3 E9.3 E9.3 COMT 28
73 ***** COMT 29
74 ***** COMT 30
75 ***** COMH 1
76 *** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS *** COMH 2
77 ***** COMH 3
78 INTEGER HMAX COMH 4
79 PARAMETER (HMAX = 35) COMH 5
80 COMMON /HOLEN/ HTOT, HERR COMH 6
81 COMMON /HOLEC/ HOLE(HMAX,4) COMH 7
82 INTEGER HTOT, HERR COMH 8
83 CHARACTER * 3 HOLE COMH 9
84 *===== COMH 10
85 * DESCRIPTION OF ARRAYS COMH 11
86 *===== COMH 12
87 * ROOM IDENTIFICATION APERTURE ID COMH 13
88 *----- COMH 14
89 * DIRECTION FROM ROOM TO ROOM COMH 15
90 *----- COMH 16
91 * HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4) COMH 17
92 * A3 A3 A3 A3 COMH 18
93 ***** COMH 19
94 ***** COMH 20
95 ***** COMR 1
96 *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS *** COMR 2
97 ***** COMR 3
98 INTEGER RMAX COMR 4
99 PARAMETER (RMAX = 20) COMR 5
100 COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR 6
101 INTEGER NROOMS COMR 7
102 REAL ROOM COMR 8
103 ***** COMR 9
104 ***** COMR 10
105 ***** COMM 1
106 *** COMMON FOR DATABASE OF MATERIAL PROPERTIES *** COMM 2
107 ***** COMM 3
108 INTEGER MMAX COMM 4
109 PARAMETER (MMAX=100) COMM 5
110 COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM 6
111 5 MFREQ(MMAX,7), MERR, MTOT COMM 7
112 COMMON /MATC/ MAT(MMAX), MATDESC(MMAX) COMM 8
113 INTEGER MTOT, MERR COMM 9
114 REAL MATTEN, MRCOEF, MFREQ, QA, QR COMM 10
115 CHARACTER * 3 MAT COMM 11
116 CHARACTER * 70 MATDESC COMM 12
117 ***** COMM 13
118 ***** COMM 14
119 ***** COMJ 1
120 * COMJ 2
121 * COMMON FOR EVALUATION OF ROOM MATRIX COMJ 3
122 * COMJ 4
123 ***** COMJ 5
124 COMMON /MAT/ TMAT(RMAX,RMAX), ENERGY(RMAX), POWER(6), FTIME COMJ 6
125 +, SWR(RMAX,6), IDIR COMJ 7
126 REAL TMAT, ENERGY, POWER, SWR COMJ 8
127 LOGICAL FTIME COMJ 9
128 ***** COMD 1

```

```

129 * COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS COMD 2
130 *****COMD 3
131 COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ),DREFL, DREFLW COMD 4
132 REAL DDABS ,DREFL , DREFLW COMD 5
133 *****COMD 6
134 *****COMD 7
135 ***** MASTER 11
136 * DECLARATION OF VARIABLES MASTER 12
137 ***** MASTER 13
138 INTEGER GETLEN MASTER 14
139 ***** MASTER 15
140 * INITIAL SETUP MASTER 16
141 ***** MASTER 17
142 NROOMS = 3 MASTER 18
143 AFLAG = 0 MASTER 19
144 RFLAG = 0 MASTER 20
145 ***** MASTER 21
146 * INPUT BUILDING IDENTIFICATION MASTER 22
147 ***** MASTER 23
148 20 PRINT*, 'ENTER BUILDING IDENTIFICATION (E.G. '101')' MASTER 24
149 PRINT*, ' (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)' MASTER 25
150 READ(1, *, END = 20) BLDG MASTER 26
151 PRINT*, 'BUILDING IDENTIFICATION ENTERED AS ''', MASTER 27
152 $ BLDG(1:GETLEN(BLDG)), '' MASTER 28
153 30 PRINT *, 'ENTER NUMBER OF ROOMS IN BUILDING' MASTER 29
154 READ (1,*,END = 30) NROOMS MASTER 30
155 ***** MASTER 31
156 * LOAD ARRAYS FROM DATA FILES MASTER 32
157 ***** MASTER 33
158 CALL LMATTER MASTER 34
159 CALL LWALL MASTER 35
160 CALL LTYPE MASTER 36
161 CALL LHOLE MASTER 37
162 CALL LFREQ MASTER 38
163 ***** MASTER 39
164 * CHECK FOR ERROR IN FREQ FILE. MASTER 40
165 * IF THERE IS AN ERROR (E.G. MISSING) THEN MASTER 41
166 * JUST USE THE DEFAULT FREQUENCIES MASTER 42
167 ***** MASTER 43
168 IF (FERR.NE.0) THEN MASTER 44
169 FTOT = 7 MASTER 45
170 DO 40 IFR = 1,7 MASTER 46
171 40 FREQA(IFR) = 1.0E03 * (10.0 ** IFR) MASTER 47
172 PRINT*, ' *** DEFAULT FREQUENCIES WILL BE USED *** ' MASTER 48
173 ENDIF MASTER 49
174 ***** MASTER 50
175 CALL PWALL MASTER 51
176 CALL PHOLE MASTER 52
177 CALL PTYPE MASTER 53
178 ***** MASTER 54
179 * CHECK FOR FILE ERROR MASTER 55
180 ***** MASTER 56
181 IF ( ( MERR .NE. 0 ) .OR. ( WERR .NE. 0 ) .OR. MASTER 57
182 $ ( HERR .NE. 0 ) .OR. ( TERR .NE. 0 ) ) THEN MASTER 58
183 CALL ERROR (5) MASTER 59
184 END IF MASTER 60
185 ***** MASTER 61
186 * MASTER 62
187 ***** MASTER 63
188 DO 200 IFR= 1,FTOT MASTER 64
189 FREQ = FREQA(IFR) MASTER 65
190 CALL IDDABS MASTER 66
191 CALL LTDB MASTER 67
192 CALL LRAREA MASTER 68

```

193	CALL CFACTOR	MASTER	69
194	CALL DFACTOR	MASTER	70
195	FTIME = .TRUE.	MASTER	71
196	DO 100 IDIR = 1 , 5	MASTER	72
197	DO 50 J = 1 , 6	MASTER	73
198	50 POWER (J) = 0.0	MASTER	74
199	POWER (IDIR) = 10.0	MASTER	75
200	CALL ECALC	MASTER	76
201	100 CALL SPWR	MASTER	77
202	CALL PPWR2	MASTER	78
203	200 CONTINUE	MASTER	79
204	STOP	MASTER	80
205	END	MASTER	81

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
DDABS	0B	/ROOMD/	REAL	676
DREFL	1244B	/ROOMD/	REAL	
DREFLW	1245B	/ROOMD/	REAL	
ENERGY	620B	/MAT/	REAL	20
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTIME	652B	/MAT/	LOGICAL	
FTOT	67B	/INITILN/	INTEGER	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
IDIR	1043B	/MAT/	INTEGER	
IFR	332B		INTEGER	
J	336B		INTEGER	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOEF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
POWER	644B	/MAT/	REAL	6
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
QUALITY	1B	/INITILN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
SWR	653B	/MAT/	REAL	120
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPECL/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TMAT	0B	/MAT/	REAL	400
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPECL/	CHAR*3	105
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME----	TYPE-----	VALUE	-NAME----	TYPE-----	VALUE
FMAX	INTEGER	50	RMAX	INTEGER	20
HMAX	INTEGER	35	TMAX	INTEGER	35
MMAX	INTEGER	100	WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----	TYPE-----	ARGS-----	CLASS-----	-NAME-----	TYPE-----	ARGS-----	CLASS-----
CFACTOR		0	SUBROUTINE	LRAREA		0	SUBROUTINE
DFACTOR		0	SUBROUTINE	LTDB		0	SUBROUTINE
ECALC		0	SUBROUTINE	LTYPE		0	SUBROUTINE
ERROR		1	SUBROUTINE	LWALL		0	SUBROUTINE
GETLEN	INTEGER	1	FUNCTION	PHOLE		0	SUBROUTINE
IDDABS		0	SUBROUTINE	PPWR2		0	SUBROUTINE
LFREQ		0	SUBROUTINE	PTYPE		0	SUBROUTINE
LHOLE		0	SUBROUTINE	PWALL		0	SUBROUTINE
LMATTER		0	SUBROUTINE	SPWR		0	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS----	PROPERTIES----	DEF	-LABEL-ADDRESS----	PROPERTIES----	DEF
20	26B	148	50	INACTIVE DO-TERM	198
30	44B	153	100	INACTIVE DO-TERM	201
40	INACTIVE DO-TERM	171	200	INACTIVE DO-TERM	203

--ENTRY POINTS--(LO=A)

-NAME---	ADDRESS--	ARGS---
MASTER	20B	0

--I/O UNITS--(LO=A)

-NAME---	PROPERTIES-----
TAPE1	FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	340B = 224
CM LABELLED COMMON LENGTH	12774B = 5628
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.183 SECONDS

[illegible]

65	EXACT = .TRUE.	ATTEN	52
66	CINDEX = C	ATTEN	53
67	ATTEN = MATTEN (RINDEX,CINDEX)	ATTEN	54
68	ATTEN = ATTEN * (1 + AFLAG / 100)	ATTEN	55
69	END IF	ATTEN	56
70 20	CONTINUE	ATTEN	57
71	*****	ATTEN	58
72 *	INTERPOLATE ATTENUATION VALUES IF EXACT FREQUENCY IS	ATTEN	59
73 *	NOT IN THE FREQUENCY/ATTENUATION ARRAYS.	ATTEN	60
74	*****	ATTEN	61
75	IF (.NOT. EXACT) THEN	ATTEN	62
76	DO 30 C=1,6	ATTEN	63
77	IF (FREQ .GT. MFREQ (RINDEX,C) .AND.	ATTEN	64
78	\$ FREQ .LT. MFREQ (RINDEX,C+1)) THEN	ATTEN	65
79	CINDEX = C	ATTEN	66
80	END IF	ATTEN	67
81 30	CONTINUE	ATTEN	68
82	F = ALOG10 (FREQ)	ATTEN	69
83	LOFREQ = ALOG10 (MFREQ (RINDEX, CINDEX))	ATTEN	70
84	HIFREQ = ALOG10 (MFREQ (RINDEX, CINDEX + 1))	ATTEN	71
85	LOATTEN = MATTEN (RINDEX,CINDEX)	ATTEN	72
86	HIATTEN = MATTEN (RINDEX, CINDEX + 1)	ATTEN	73
87	FRAC = (F - LOFREQ) / (HIFREQ - LOFREQ)	ATTEN	74
88	ATTEN = LOATTEN + (FRAC * (HIATTEN - LOATTEN))	ATTEN	75
89	ATTEN = ATTEN * (1 + AFLAG / 100)	ATTEN	76
90	END IF	ATTEN	77
91	RETURN	ATTEN	78
92	END	ATTEN	79

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	3	DUMMY-ARG	REAL	
ATTEN	242B		REAL	
C	254B		INTEGER	
CINDEX	256B		INTEGER	
EXACT	260B		LOGICAL	
F	252B		REAL	
FOUND	257B		LOGICAL	
FRAC	243B		REAL	
FREQ	2	DUMMY-ARG	REAL	
HIATTEN	251B		REAL	
HIFREQ	247B		REAL	
ID	1	DUMMY-ARG	CHAR*3	
IERR	262B		INTEGER	
LOATTEN	250B		REAL	
LOFREQ	246B		REAL	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATTEN	0B	/MATN/	REAL	700
MAXFREQ	245B		REAL	
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MINFREQ	244B		REAL	
MRCOEF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	253B		INTEGER	
RINDEX	255B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)
-NAME---TYPE-----VALUE

MMAX INTEGER 100

--PROCEDURES--(LO=A)
-NAME-----TYPE-----ARGS-----CLASS-----

ALOG10 REAL 1 INTRINSIC
ERROR 1 SUBROUTINE

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES---DEF

10 INACTIVE DO-TERM 44
20 INACTIVE DO-TERM 70
30 INACTIVE DO-TERM 81

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS---ARGS---

ATTN 6B 3

--STATISTICS--

PROGRAM-UNIT LENGTH 267B = 183
CM LABELLED COMMON LENGTH 5730B = 3032
CM STORAGE USED 61000B = 25088
COMPILE TIME 0.132 SECONDS

1	SUBROUTINE ERROR(IERR)	ERROR	1
2	CHARACTER*45 MESSAGE(20)	ERROR	2
3	DATA MESSAGE(1)/'MATERIALS DATA BASE IS EMPTY	' / ERROR	3
4	DATA MESSAGE(2)/'FREQUENCY IS OUT OF RANGE	' / ERROR	4
5	DATA MESSAGE(3)/'THIS MATERIAL IS NOT IN DATA BASE	' / ERROR	5
6	DATA MESSAGE(4)/'DENOMINATOR IS ZERO	' / ERROR	6
7	DATA MESSAGE(5)/'FILE HANDLING ERROR	' / ERROR	7
8	DATA MESSAGE(6)/'ERROR CODE IS OUT OF RANGE	' / ERROR	8
9	DATA MESSAGE(7)/'ERROR CODE IS OUT OF RANGE	' / ERROR	9
10	DATA MESSAGE(8)/'ERROR CODE IS OUT OF RANGE	' / ERROR	10
11	DATA MESSAGE(9)/'ERROR CODE IS OUT OF RANGE	' / ERROR	11
12	DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE	' / ERROR	12
13	DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE	' / ERROR	13
14	DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE	' / ERROR	14
15	DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE	' / ERROR	15
16	DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE	' / ERROR	16
17	DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE	' / ERROR	17
18	DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE	' / ERROR	18
19	DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE	' / ERROR	19
20	DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE	' / ERROR	20
21	DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE	' / ERROR	21
22	DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE	' / ERROR	22
23	IERRM=5	ERROR	23
24	IF(IERR.GT.IERRM) IERR=20	ERROR	24
25	WRITE(6,10) IERR,MESSAGE(IERR)	ERROR	25
26	10 FORMAT(' ***ERROR NUMBER = ',I5,' *** ',A45)	ERROR	26
27	CALL PMDSTOP	ERROR	27
28	STOP 'ERROR'	ERROR	28
29	END	ERROR	29

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

IERR	1	DUMMY-ARG	INTEGER	
IERRM	210B		INTEGER	
MESSAGE	56B		CHAR*45	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PMDSTOP		0	SUBROUTINE
---------	--	---	------------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	36B	FORMAT	26
----	-----	--------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

ERROR	5B	1
-------	----	---

FTN 5 1+552 84/03/14. 10.18.23 PAGE 10
SUBROUTINE ERROR 74/175 OPT=0

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	213B = 139
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.055 SECONDS

1	SUBROUTINE CFACTOR	CFACTOR	1
2	*[[[CFACTOR	2
3	*[[[CFACTOR	3
4	*[[[THIS ROUTINE CALCULATES THE ATTENUATION OF EACH WALL AND EACH	CFACTOR	4
5	*[[[OPENING IN EACH WALL, LAYER BY LAYER, AND THEN CALCULATES THE	CFACTOR	5
6	*[[[COMPOSITE TRANSMISSION FACTORS OF EACH WALL USING AN APPROACH	CFACTOR	6
7	*[[[DEVELOPED BY JERRY WYSS.	CFACTOR	7
8	*[[[CFACTOR	8
9	*[[[CFACTOR	9
10	*****CFACTOR	CFACTOR	10
11	*** VARIABLE DEFINITIONS:	CFACTOR	11
12	*** WATTEN: WALL ATTENUATION	CFACTOR	12
13	*** OATTEN: OPENING ATTENUATION	CFACTOR	13
14	*** LATTEN: LAYER ATTENUATION	CFACTOR	14
15	*** MATTEN: MATERIAL ATTENUATION	CFACTOR	15
16	*** MAT: MATERIAL IDENTIFICATION	CFACTOR	16
17	*** ID: IDENTIFICATION OF OPENING	CFACTOR	17
18	*** WALL: WALL ARRAY CONTAINING WALL IDENTIFICATION AND MATERIAL	CFACTOR	18
19	*** WDIM: WALL ARRAY CONTAINING PHYSICAL DIMENSIONS OF THE WALL	CFACTOR	19
20	*** WMAX: MAXIMUM SIZE OF WALL AND WDIM ARRAYS	CFACTOR	20
21	*** WTOT: TOTAL LINES OF DATA IN THE THE WALL AND WDIM ARRAYS.	CFACTOR	21
22	*** HEIGHT: HEIGHT OF WALL	CFACTOR	22
23	*** WIDTH: WIDTH OF WALL	CFACTOR	23
24	*** WDIM(R,C): THICKNESS OF WALL	CFACTOR	24
25	*** AREA: AREA	CFACTOR	25
26	*** WAREA: TOTAL WALL AREA WITHOUT SUBTRACTING OPENINGS.	CFACTOR	26
27	*** OAREA: TOTAL AREA OF THE OPENINGS.	CFACTOR	27
28	*** NEWWALL: TRUE IF DATA LINE BELONGS TO A NEW WALL	CFACTOR	28
29	*** WALLEND: TRUE IF DATA LINE IS THE LAST DATA LINE OF A WALL	CFACTOR	29
30	*** T: TRANSMISSION FACTOR	CFACTOR	30
31	*** S: AREA OF OPENING OR WALL AS APPROPRIATE	CFACTOR	31
32	*** TS: TRANSMISSION FACTOR = T1*S1 + T2*S2 + T3*S3 + ...	CFACTOR	32
33	*** TS2: TRANSMISSION FACTOR = T1*S1*S1 + T2*S2*S2 + ...	CFACTOR	33
34	*****CFACTOR	CFACTOR	34
35	*****COMF	COMF	1
36	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
37	*****COMF	COMF	3
38	INTEGER FMAX	COMF	4
39	PARAMETER (FMAX = 50)	COMF	5
40	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
41	\$ FTOT	COMF	7
42	COMMON /INITILC/ BLDG	COMF	8
43	CHARACTER * 5 BLDG	COMF	9
44	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
45	INTEGER QUALITY, FERR, FTOT	COMF	11
46	*****COMF	COMF	12
47	*****COMF	COMF	13
48	*****COMH	COMH	1
49	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS	***COMH	2
50	*****COMH	COMH	3
51	INTEGER HMAX	COMH	4
52	PARAMETER (HMAX = 35)	COMH	5
53	COMMON /HOLEN/ HTOT, HERR	COMH	6
54	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
55	INTEGER HTOT, HERR	COMH	8
56	CHARACTER * 3 HOLE	COMH	9
57	* =====	COMH	10
58	* DESCRIPTION OF ARRAYS	COMH	11
59	* =====	COMH	12
60	* ROOM IDENTIFICATION APERTURE ID	COMH	13
61	* -----	COMH	14
62	* DIRECTION FROM ROOM TO ROOM	COMH	15
63	* -----	COMH	16
64	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17

```

65 *      A3              A3              A3              A3              COMH      18
66 *****COMH      19
67 *****COMH      20
68 *****COMT      1
69 ***  COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS      ***COMT      2
70 *****COMT      3
71      INTEGER TMAX      COMT      4
72      PARAMETER (TMAX=35)      COMT      5
73      COMMON /TYPEN/ TDIM(TMAX,4), TTOT, TDB2(TMAX,2), TDBTOT, TERR      COMT      6
74      COMMON /TYPEC/ TYPE(TMAX,3), TDB1(TMAX)      COMT      7
75      INTEGER TTOT, TDBTOT, TERR      COMT      8
76      REAL TDIM, TDB2      COMT      9
77      CHARACTER * 3 TYPE, TDB1      COMT     10
78 *=====COMT     11
79 * DESCRIPTION OF ARRAYS      COMT     12
80 *=====COMT     13
81 *      ID      MATERIAL      FRAME MATERIAL      COMT     14
82 *-----COMT     15
83 *TYPE(X,1)      TYPE(X,2)      TYPE(X,3)      COMT     16
84 * A3      A3      A3      COMT     17
85 *=====COMT     18
86 * HEIGHT      WIDTH      LAYER      DISTANCE      COMT     19
87 *      THICKNESS      ABOVE FLOOR      COMT     20
88 *-----COMT     21
89 * TDIM(X,1)      TDIM(X,2)      TDIM(X,3)      TDIM(X,4)      COMT     22
90 * F8.2      F8.2      F8.2      F8.2      COMT     23
91 *-----COMT     24
92 * ID      ATTENUATION      AREA      COMT     25
93 *-----COMT     26
94 * TDB1(X)      TDB2(X,1)      TDB2(X,2)      COMT     27
95 * A3      E9.3      E9.3      COMT     28
96 *****COMT     29
97 *****COMT     30
98 *****COMR      1
99 ***  COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS      ***COMR      2
100 *****COMR      3
101      INTEGER RMAX      COMR      4
102      PARAMETER (RMAX = 20)      COMR      5
103      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR      6
104      INTEGER NROOMS      COMR      7
105      REAL ROOM      COMR      8
106 *****COMR      9
107 *****COMR     10
108 *****COMW      1
109 ***  COMMON FOR DATABASE OF WALL PARAMETERS      ***COMW      2
110 *****COMW      3
111      INTEGER WMAX      COMW      4
112      PARAMETER (WMAX = 75)      COMW      5
113      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR      COMW      6
114      COMMON /WALLC/ WALL(WMAX,4)      COMW      7
115      INTEGER WTOT, WERR      COMW      8
116      REAL WDIM      COMW      9
117      CHARACTER *3 WALL      COMW     10
118 *=====COMW     11
119 ** DESCRIPTION OF ARRAYS      COMW     12
120 *=====COMW     13
121 *      WALL IDENTIFICATION      COMW     14
122 *-----COMW     15
123 * DIRECTION      FROM      TO      COMW     16
124 *      ROOM      ROOM      COMW     17
125 *-----COMW     18
126 * WALL(X,1)      WALL(X,2)      WALL(X,3)      COMW     19
127 * A3      A3      A3      COMW     20
128 *=====COMW     21

```

129 *	WALL PARAMETERS	COMW	22
130 *	-----	COMW	23
131 *	MATERIAL HEIGHT WIDTH LAYER THICKNESS	COMW	24
132 *	-----	COMW	25
133 *	WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3)	COMW	26
134 *	A3 F8.2 F8.2 F8.2	COMW	27
135	*****	COMW	28
136	*****	COMW	29
137	*****	COMD	1
138 *	COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS	COMD	2
139	*****	COMD	3
140	COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6) ,DREFL, DREFLW	COMD	4
141	REAL DDABS ,DREFL , DREFLW	COMD	5
142	*****	COMD	6
143	*****	COMD	7
144	*****	CFACTOR	41
145 *	DECLARATION OF VARIABLES	CFACTOR	42
146	*****	CFACTOR	43
147	INTEGER NEXT, LAST, R, C, ROW	CFACTOR	44
148	REAL WATTEN,LATTEN,OATTEN,MATTEN,ATTEN,T,S,TS,TS2	CFACTOR	45
149	REAL HEIGHT,WIDTH,AREA,OAREA,WAREA	CFACTOR	46
150	CHARACTER * 3 FROM, TO, MAT, ID	CFACTOR	47
151	LOGICAL NEWWALL,WALLEND	CFACTOR	48
152	*****	CFACTOR	49
153 *	INITIALIZE ROOM MATRIX.	CFACTOR	50
154	*****	CFACTOR	51
155	DO 5 R = 1,RMAX	CFACTOR	52
156	DO 5 C = 1,RMAX	CFACTOR	53
157	ROOM(R,C) = 0.0	CFACTOR	54
158	5 CONTINUE	CFACTOR	55
159	DO 6 R = 1,RMAX	CFACTOR	56
160	DO 6 C = RMAX + 1, RMAX + 5	CFACTOR	57
161	ROOM(R,C) = 0.0	CFACTOR	58
162	6 CONTINUE	CFACTOR	59
163	*****	CFACTOR	60
164 *	LOOP & PROCESS EACH LAYER OF THE WALL ARRAY	CFACTOR	61
165	*****	CFACTOR	62
166	DO 10 R = 1,WTOT	CFACTOR	63
167	*****	CFACTOR	64
168 *	SET WALLEND CONDITION	CFACTOR	65
169	*****	CFACTOR	66
170	NEXT = R + 1	CFACTOR	67
171	IF (R .EQ. WTOT) THEN	CFACTOR	68
172	WALLEND = .TRUE.	CFACTOR	69
173	ELSE IF (WALL(R,2) .NE. WALL(NEXT,2) .OR.	CFACTOR	70
174	\$ WALL(R,3) .NE. WALL(NEXT,3)) THEN	CFACTOR	71
175	WALLEND = .TRUE.	CFACTOR	72
176	ELSE	CFACTOR	73
177	WALLEND = .FALSE.	CFACTOR	74
178	END IF	CFACTOR	75
179	*****	CFACTOR	76
180 *	SET NEWWALL CONDITION	CFACTOR	77
181	*****	CFACTOR	78
182	LAST = R - 1	CFACTOR	79
183	IF (R .EQ. 1) THEN	CFACTOR	80
184	NEWWALL = .TRUE.	CFACTOR	81
185	ELSE IF (WALL(R,2) .NE. WALL(LAST,2) .OR.	CFACTOR	82
186	\$ WALL(R,3) .NE. WALL(LAST,3)) THEN	CFACTOR	83
187	NEWWALL = .TRUE.	CFACTOR	84
188	ELSE	CFACTOR	85
189	NEWWALL = .FALSE.	CFACTOR	86
190	END IF	CFACTOR	87
191	*****	CFACTOR	88
192	*****	CFACTOR	89

193 * CALCULATE WALL ATTENUATION, LAYER BY LAYER	CFACTOR	90
194 *****	CFACTOR	91
195 IF (NEWWALL) THEN	CFACTOR	92
196 *****	CFACTOR	93
197 * INITIALIZE WALL CONDITIONS	CFACTOR	94
198 *****	CFACTOR	95
199 TS = 0	CFACTOR	96
200 TS2 = 0	CFACTOR	97
201 WATTEN = 0	CFACTOR	98
202 END IF	CFACTOR	99
203 *****	CFACTOR	100
204 * CALCULATE ATTENUATION OF LAYER	CFACTOR	101
205 *****	CFACTOR	102
206 MAT = WALL(R,4)	CFACTOR	103
207 MATTEN = ATTEN (MAT,FREQ,AFLAG)	CFACTOR	104
208 LATTEN = MATTEN * WDIM(R,3)	CFACTOR	105
209 *****	CFACTOR	106
210 * ACCUMULATE ATTENUATION OF WALL FROM LAYERS	CFACTOR	107
211 *****	CFACTOR	108
212 WATTEN = WATTEN + LATTEN	CFACTOR	109
213 *****	CFACTOR	110
214 *****	CFACTOR	111
215 * CHECK IF END-OF-WALL LAYER	CFACTOR	112
216 * AND THEN CALCULATE VALUES FOR HOLES IF TRUE.	CFACTOR	113
217 * OTHERWISE GO BACK AND DO THE NEXT LAYER.	CFACTOR	114
218 *****	CFACTOR	115
219 IF (WALLEND) THEN	CFACTOR	116
220 FROM = WALL(R,2)	CFACTOR	117
221 TO = WALL(R,3)	CFACTOR	118
222 *****	CFACTOR	119
223 * CALCULATE ATTENUATION OF OPENINGS	CFACTOR	120
224 * ...AND TOTAL AREA OF OPENINGS	CFACTOR	121
225 *****	CFACTOR	122
226 OAREA = 0	CFACTOR	123
227 DO 20 ROW = 1, HTOT	CFACTOR	124
228 *****	CFACTOR	125
229 * ...CHECK FOR A HOLE IN PRESENT WALL	CFACTOR	126
230 *****	CFACTOR	127
231 IF (HOLE(ROW,2) .EQ. FROM .AND. HOLE(ROW,3) .EQ. TO) THEN	CFACTOR	128
232 *****	CFACTOR	129
233 * ...IF THERE IS A MATCH, CALCULATE ITS CONTRIBUSION;	CFACTOR	130
234 * ...OTHERWISE KEEP SEARCHING HOLE'S TABLE	CFACTOR	131
235 *****	CFACTOR	132
236 ID = HOLE(ROW,4)	CFACTOR	133
237 *****	CFACTOR	134
238 * ...GET ATTENUATION AND AREA OF HOLE	CFACTOR	135
239 *****	CFACTOR	136
240 CALL SRCHTDB(ID, OATTEN,AREA)	CFACTOR	137
241 OAREA = OAREA + AREA	CFACTOR	138
242 CALL RESOND (ID)	CFACTOR	139
243 IF (OATTEN .LE. 120) THEN	CFACTOR	140
244 *****	CFACTOR	141
245 * ...CALCULATE TRANSMISSION OF HOLE.	CFACTOR	142
246 * ...SET TO ZERO IF LESS THAN 120 DB	CFACTOR	143
247 *****	CFACTOR	144
248 T = 10*((-OATTEN + DREFLW) / 10)	CFACTOR	145
249 ELSE	CFACTOR	146
250 T = 0	CFACTOR	147
251 END IF	CFACTOR	148
252 S = AREA	CFACTOR	149
253 *****	CFACTOR	150
254 * ...ACCUMULATE TRANSMISSION * AREA AND	CFACTOR	151
255 * ...TRANSMISSION * AREA * AREA	CFACTOR	152
256 * ...FOR HOLES IN WALL.	CFACTOR	153

257	*****	CFACTOR	154
258	TS = TS + T * S	CFACTOR	155
259	TS2 = TS2 + T * S * S	CFACTOR	156
260	END IF	CFACTOR	157
261	20 CONTINUE	CFACTOR	158
262	*****	CFACTOR	159
263	*****	CFACTOR	160
264	*****	CFACTOR	161
265	* CALCULATE & STORE ATTENUATION OF EACH ROOM	CFACTOR	162
266	*****	CFACTOR	163
267	* ...CALCULATE TOTAL WALL AREA	CFACTOR	164
268	*****	CFACTOR	165
269	HEIGHT = WDIM(R,1)	CFACTOR	166
270	WIDTH = WDIM(R,2)	CFACTOR	167
271	WAREA = HEIGHT * WIDTH	CFACTOR	168
272	S = WAREA - OAREA	CFACTOR	169
273	*****	CFACTOR	170
274	* ...CALCULATE ATTENUATION	CFACTOR	171
275	*****	CFACTOR	172
276	IF (WATTEN .LE. 120.) THEN	CFACTOR	173
277	*****	CFACTOR	174
278	* ...CALCULATE TRANSMISSION OF WALL.	CFACTOR	175
279	* ...SET TO ZERO IF LESS THAN -120 DB.	CFACTOR	176
280	*****	CFACTOR	177
281	T = 10**(-WATTEN / 10)	CFACTOR	178
282	ELSE	CFACTOR	179
283	T = 0	CFACTOR	180
284	END IF	CFACTOR	181
285	TS = TS + T * S	CFACTOR	182
286	TS2 = TS2 + T * S * S	CFACTOR	183
287	*****	CFACTOR	184
288	* ...INSERT TOTAL TRANSMISSION * AREA OF WALL INTO ROOM ARRAY	CFACTOR	185
289	*****	CFACTOR	186
290	CALL LROOM (TS,TS2,FROM,TO)	CFACTOR	187
291	END IF	CFACTOR	188
292	10 CONTINUE	CFACTOR	189
293	RETURN	CFACTOR	190
294	END	CFACTOR	191

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
AREA	500B		REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	464B		INTEGER	
DDABS	0B	/ROOMD/	REAL	676
DREFL	1244B	/ROOMD/	REAL	
DREFLW	1245B	/ROOMD/	REAL	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FROM	503B		CHAR*3	
FTOT	67B	/INITILN/	INTEGER	
HEIGHT	476B		REAL	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
ID	506B		CHAR*3	
LAST	462B		INTEGER	
LATTEN	467B		REAL	
MAT	505B		CHAR*3	
MATTEN	471B		REAL	

NEWALL	507B		LOGICAL	
NEXT	461B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
OAREA	501B		REAL	
OATTEN	470B		REAL	
QUALITY	1B	/INITILN/	INTEGER	
R	463B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
ROW	465B		INTEGER	
S	473B		REAL	
T	472B		REAL	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPECL/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TO	504B		CHAR*3	
TS	474B		REAL	
TS2	475B		REAL	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPECL/	CHAR*3	105
WALL	0B	/WALLCL/	CHAR*3	300
WALLEND	510B		LOGICAL	
WAREA	502B		REAL	
WATTEN	466B		REAL	
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WIDTH	477B		REAL	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
HMAX	INTEGER	35
RMAX	INTEGER	20
TMAX	INTEGER	35
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

ATTEN	REAL	3	FUNCTION
LROOM		4	SUBROUTINE
RESOND		1	SUBROUTINE
SRCHTDB		3	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

5	INACTIVE	DO-TERM	158
6	INACTIVE	DO-TERM	162
10	INACTIVE	DO-TERM	292
20	INACTIVE	DO-TERM	261

FTN 5.1+552 84/03/14. 10.18.23 PAGE 17
SUBROUTINE CFACTOR 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS---ARGS---

CFACTOR 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH	521B = 337
CM LABELLED COMMON LENGTH	4000B = 2048
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.249 SECONDS

1	SUBROUTINE LHOLE	LHOLE	1
2	*[[[[[[LHOLE	2
3	*[[[[[[LHOLE	3
4	*[[[LOAD THE CONTENTS OF THE "HOLE" FILE INTO THE "HOLE" ARRAY	[[[LHOLE	4
5	*[[[[[[LHOLE	5
6	*[[[[[[LHOLE	6
7	*****COMH		1
8	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS ***	COMH	2
9	*****COMH		3
10	INTEGER HMAX	COMH	4
11	PARAMETER (HMAX = 35)	COMH	5
12	COMMON /HOLEN/ HTOT, HERR	COMH	6
13	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
14	INTEGER HTOT, HERR	COMH	8
15	CHARACTER * 3 HOLE	COMH	9
16	* =====	COMH	10
17	* DESCRIPTION OF ARRAYS	COMH	11
18	* =====	COMH	12
19	* ROOM IDENTIFICATION APERTURE ID	COMH	13
20	* -----	COMH	14
21	* DIRECTION FROM ROOM TO ROOM	COMH	15
22	* -----	COMH	16
23	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17
24	* A3 A3 A3 A3	COMH	18
25	*****COMH		19
26	*****COMH		20
27	*****COMF		1
28	*** COMMON FOR INITIAL PARAMETERS ***	COMF	2
29	*****COMF		3
30	INTEGER FMAX	COMF	4
31	PARAMETER (FMAX = 50)	COMF	5
32	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
33	\$ FTOT	COMF	7
34	COMMON /INITILC/ BLDG	COMF	8
35	CHARACTER * 5 BLDG	COMF	9
36	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
37	INTEGER QUALITY, FERR, FTOT	COMF	11
38	*****COMF		12
39	*****COMF		13
40	INTEGER GETLEN, R, C	LHOLE	9
41	CHARACTER * 7 PFN	LHOLE	10
42	PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'H'	LHOLE	11
43	HERR = 0	LHOLE	12
44	CALL PF ('GET',0,PFN(1:GETLEN(PFN)), 'RC',HERR)	LHOLE	13
45	IF (HERR .EQ. 0) THEN	LHOLE	14
46	OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',	LHOLE	15
47	\$ STATUS='OLD', ACCESS='SEQUENTIAL')	LHOLE	16
48	1000 FORMAT (1X,4(1X,A3))	LHOLE	17
49	HTOT = 0	LHOLE	18
50	DO 10 R = 1,HMAX	LHOLE	19
51	READ (3,1000,END=20)(HOLE(R,C),C=1,4)	LHOLE	20
52	HTOT = HTOT + 1	LHOLE	21
53	10 CONTINUE	LHOLE	22
54	20 CONTINUE	LHOLE	23
55	CLOSE(3,STATUS='DELETE')	LHOLE	24
56	ELSE IF (HERR .EQ. 2) THEN	LHOLE	25
57	CALL WARNING (1)	LHOLE	26
58	ELSE	LHOLE	27
59	CALL WARNING (2)	LHOLE	28
60	END IF	LHOLE	29
61	RETURN	LHOLE	30
62	END	LHOLE	31

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	214B		INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
PFN	215B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
R	213B		INTEGER	
RFLAG	3B	/INITILN/	REAL	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
HMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	53
20	73B		54
1000	130B	FORMAT	48

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

LHOLE	5B	0
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE3	AUX/FMT/SEQ
-------	-------------

--STATISTICS--

PROGRAM-UNIT LENGTH	222B = 146
CM LABELLED COMMON LENGTH	145B = 101
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.094 SECONDS

[illegible]

65	R = VAL(MATID(2:3))	LMATTER	52
66	MAT (R) = MATID	LMATTER	53
67	READ (3,2000,END=20) MATDESC (R)	LMATTER	54
68	READ (3,4000,END=20) (MFREQ(R,C),C=1,7)	LMATTER	55
69	READ (3,4000,END=20) (MATTEN(R,C),C=1,7)	LMATTER	56
70	READ (3,4000,END=20) QA (R)	LMATTER	57
71	READ (3,4000,END=20) (MRCOE(R,C),C=1,7)	LMATTER	58
72	READ (3,4000,END=20) QR (R)	LMATTER	59
73	GOTO 10	LMATTER	60
74 20	CONTINUE	LMATTER	61
75 1000	FORMAT (A3)	LMATTER	62
76 2000	FORMAT (A70)	LMATTER	63
77 4000	FORMAT (7(1X,E9.3))	LMATTER	64
78	*****	LMATTER	65
79 *	CLOSE FILE	LMATTER	66
80	*****	LMATTER	67
81	CLOSE (3,STATUS = 'DELETE')	LMATTER	68
82	RETURN	LMATTER	69
83	END	LMATTER	70

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	327B		INTEGER	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	330B		CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOE	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	326B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

MMAX	INTEGER	100
------	---------	-----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

PF		5	SUBROUTINE
VAL	INTEGER	1	FUNCTION
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

10	36B	64	1000	202B	FORMAT	75
20	160B	74	2000	204B	FORMAT	76
999	*NO REFS*	37	4000	206B	FORMAT	77

FTN 5 1+552 84/03/14. 10.18.23 PAGE 22
SUBROUTINE LMATTER 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

LMATTER 5B 0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE3 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	336B = 222
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.138 SECONDS

```

1 SUBROUTINE LRAREA                                LRAREA          1
2 *[[[[]]]]LRRAREA                               LRRAREA          2
3 *[[[[]]]]LRRAREA                               LRRAREA          3
4 *[[[ CALCULATE THE SURFACE AREA OF EACH ROOM   LRRAREA          4
5 *[[[ AND INSERT IT IN THE "RAREA" ARRAY        LRRAREA          5
6 *[[[[]]]]LRRAREA                               LRRAREA          6
7 *[[[[]]]]LRRAREA                               LRRAREA          7
8 *****COMR                                     COMR           1
9 *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR       2
10 *****COMR                                    COMR           3
11 INTEGER RMAX                                       COMR           4
12 PARAMETER (RMAX = 20)                             COMR           5
13 COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR           6
14 INTEGER NROOMS                                      COMR           7
15 REAL ROOM                                           COMR           8
16 *****COMR                                     COMR           9
17 *****COMR                                     COMR          10
18 *****COMW                                     COMW           1
19 *** COMMON FOR DATABASE OF WALL PARAMETERS         **COMW          2
20 *****COMW                                     COMW           3
21 INTEGER WMAX                                       COMW           4
22 PARAMETER (WMAX = 75)                             COMW           5
23 COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR            COMW           6
24 COMMON /WALLC/ WALL(WMAX,4)                       COMW           7
25 INTEGER WTOT,WERR                                  COMW           8
26 REAL WDIM                                          COMW           9
27 CHARACTER *3 WALL                                 COMW          10
28 * =====                                         COMW          11
29 ** DESCRIPTION OF ARRAYS                          COMW          12
30 * =====                                         COMW          13
31 * WALL IDENTIFICATION                            COMW          14
32 * -----                                         COMW          15
33 * DIRECTION      FROM      TO                    COMW          16
34 *              ROOM      ROOM                     COMW          17
35 * -----                                         COMW          18
36 * WALL(X,1)     WALL(X,2)    WALL(X,3)           COMW          19
37 * A3             A3           A3                   COMW          20
38 * =====                                         COMW          21
39 * WALL PARAMETERS                                   COMW          22
40 * -----                                         COMW          23
41 * MATERIAL      HEIGHT     WIDTH      LAYER THICKNESS COMW          24
42 * -----                                         COMW          25
43 * WALL(X,4)     WDIM(X,1)   WDIM(X,2)   WDIM(X,3)   COMW          26
44 * A3             F8.2       F8.2       F8.2         COMW          27
45 *****COMW                                     COMW          28
46 *****COMW                                     COMW          29
47 *****LRAREA                                LRAREA          10
48 * DECLARATION OF VARIABLES                      LRAREA          11
49 *****LRAREA                                LRAREA          12
50 INTEGER NEXT, LAST, R, RNUM, VAL                LRAREA          13
51 CHARACTER * 3 FROM, TO                           LRAREA          14
52 LOGICAL NEWWALL, WALLEND                        LRAREA          15
53 *****LRAREA                                LRAREA          16
54 * LRAREA                                        LRAREA          17
55 *****LRAREA                                LRAREA          18
56 *****LRAREA                                LRAREA          19
57 * INITIALIZE RAREA MATRIX TO ZERO               LRAREA          20
58 *****LRAREA                                LRAREA          21
59 DO 5 I = 1,RMAX                                  LRAREA          22
60 5 RAREA(I) = 0.0                                  LRAREA          23
61 *****LRAREA                                LRAREA          24
62 * LRAREA                                        LRAREA          25
63 *****LRAREA                                LRAREA          26
64 DO 10 R = 1.WTOT                                  LRAREA          27

```

65 *****	LRAREA	28
66 * SET WALLEND CONDITION	LRAREA	29
67 *****	LRAREA	30
68 NEXT = R + 1	LRAREA	31
69 IF (R .EQ. WTOT) THEN	LRAREA	32
70 WALLEND = .TRUE.	LRAREA	33
71 ELSE IF (WALL(R,2) .NE. WALL(NEXT,2) .OR.	LRAREA	34
72 \$ WALL(R,3) .NE. WALL(NEXT,3)) THEN	LRAREA	35
73 WALLEND = .TRUE.	LRAREA	36
74 ELSE	LRAREA	37
75 WALLEND = .FALSE.	LRAREA	38
76 END IF	LRAREA	39
77 *****	LRAREA	40
78 * SET NEWWALL CONDITION	LRAREA	41
79 *****	LRAREA	42
80 LAST = R - 1	LRAREA	43
81 IF (R .EQ. 1) THEN	LRAREA	44
82 NEWWALL = .TRUE.	LRAREA	45
83 ELSE IF (WALL(R,2) .NE. WALL(LAST,2) .OR.	LRAREA	46
84 \$ WALL(R,3) .NE. WALL(LAST,3)) THEN	LRAREA	47
85 NEWWALL = .TRUE.	LRAREA	48
86 ELSE	LRAREA	49
87 NEWWALL = .FALSE.	LRAREA	50
88 END IF	LRAREA	51
89 *****	LRAREA	52
90 * INSERT THE AREA INTO THE ARRAY	LRAREA	53
91 *****	LRAREA	54
92 IF (NEWWALL) THEN	LRAREA	55
93 FROM = WALL (R,2)	LRAREA	56
94 TO = WALL (R,3)	LRAREA	57
95 IF (FROM(1:1) .EQ. 'D') THEN	LRAREA	58
96 RNUM = VAL (TO(1:2))	LRAREA	59
97 RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)	LRAREA	60
98 ELSE IF (TO(1:1) .EQ. 'D') THEN	LRAREA	61
99 RNUM = VAL (FROM(1:2))	LRAREA	62
100 RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)	LRAREA	63
101 ELSE IF ((FROM(1:1) .NE. 'D') .AND. (TO(1:1) .NE. 'D')) THEN	LRAREA	64
102 RNUM = VAL (FROM(1:2))	LRAREA	65
103 RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)	LRAREA	66
104 RNUM = VAL (TO(1:2))	LRAREA	67
105 RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)	LRAREA	68
106 END IF	LRAREA	69
107 END IF	LRAREA	70
108 *	LRAREA	71
109 10 CONTINUE	LRAREA	72
110 RETURN	LRAREA	73
111 END	LRAREA	74

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

FROM	275B		CHAR*3	
I	301B		INTEGER	
LAST	272B		INTEGER	
NEWWALL	277B		LOGICAL	
NEXT	271B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
R	273B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RNUM	274B		INTEGER	
ROOM	0B	/ROOMN/	REAL	676
TO	276B		CHAR*3	
WALL	0B	/WALLC/	CHAR*3	300

WALLEND	300B		LOGICAL	
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

RMAX	INTEGER	20
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

VAL	INTEGER	1	FUNCTION
-----	---------	---	----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

5	INACTIVE	DO-TERM	60
10	INACTIVE	DO-TERM	109

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

LRAREA	5B	0
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	306B = 198
CM LABELLED COMMON LENGTH	1766B = 1014
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.133 SECONDS

```

1 SUBROUTINE LROOM (TS,TS2,FROM,TO )                                1
2 *[[[[]]]]LROOM                                                    2
3 *[[[[]]]]LROOM                                                    3
4 *** THIS ROUTINE LOADS THE TRANSMISSION COEFFICIENT INTO THE APPROPRIATELROOM 4
5 *** LOCATION IN THE 'ROOM' ARRAY.                                  LROOM 5
6 ***                                                                    LROOM 6
7 *** NROOMS: TOTAL NUMBERS OF ROOMS REPRESENTED BY DATA          LROOM 7
8 *** RMAX: MAXIMUM NUMBER POSSIBLE UNDER THE PRESENT PROGRAM CONFIGURATIOLOOM 8
9 *** TS AND TS2: TRANSMISSION COEFFICIENTS                          LROOM 9
10 *** FROM: TO: CONTAINS ROOM#'S OR THE DIRECTIONS D1,D2,4,D5,OR D6.    LROOM 10
11 *[[[[]]]]LROOM                                                  11
12 *[[[[]]]]LROOM                                                  12
13 *****LROOM                                                    13
14 *****COMR                                                       1
15 *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR 2
16 *****COMR                                                       3
17 INTEGER RMAX                                                         COMR 4
18 PARAMETER (RMAX = 20)                                                COMR 5
19 COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR 6
20 INTEGER NROOMS                                                        COMR 7
21 REAL ROOM                                                            COMR 8
22 *****COMR                                                       9
23 *****COMR                                                      10
24 *****LROOM                                                     15
25 * DECLARATION OF VARIABLES                                          LROOM 16
26 *****LROOM                                                     17
27 INTEGER VAL, C, R, RNUM, D                                           LROOM 18
28 REAL TS, TS2                                                          LROOM 19
29 CHARACTER * 3 FROM, TO                                              LROOM 20
30 *****LROOM                                                     21
31 *                                                                      LROOM 22
32 *****LROOM                                                     23
33 IF ( FROM(1:1) .EQ. 'D' ) THEN                                       LROOM 24
34     RNUM = VAL ( TO(1:2) )                                           LROOM 25
35     D = VAL ( FROM(2:2) )                                            LROOM 26
36 *****LROOM                                                     27
37 * INSERT TRANSMISSION COEFFICIENT FOR ENERGY ENTERING A ROOM FROM THE LROOM 28
38 * OUTSIDE OF THE BUILDING.                                           LROOM 29
39 *****LROOM                                                     30
40     R = NROOMS + D                                                   LROOM 31
41     C = RNUM                                                           LROOM 32
42     ROOM(R,C) = TS + ROOM(R,C)                                        LROOM 33
43 *****LROOM                                                     34
44 * INSERT TRANSMISSION COEFFICIENT INTO 'ROOM' ARRAY FOR ENERGY LEAVING LROOM 35
45 * A ROOM TO THE OUTSIDE OF THE BUILDING.                             LROOM 36
46 *****LROOM                                                     37
47     R = RNUM                                                           LROOM 38
48     C = NROOMS + D                                                    LROOM 39
49     ROOM(R,C) = TS2 / RAREA(RNUM) + ROOM(R,C)                       LROOM 40
50 *****LROOM                                                     41
51 *                                                                      LROOM 42
52 *****LROOM                                                     43
53 ELSE IF ( TO(1:1) .EQ. 'D' ) THEN                                    LROOM 44
54     RNUM = VAL ( FROM(1:2) )                                         LROOM 45
55     D = VAL ( TO(2:2) )                                              LROOM 46
56 *****LROOM                                                     47
57 * INSERT TRANSMISSION COEFFICIENT INTO 'ROOM' ARRAY FOR ENERGY ENTERINGLOOM 48
58 * A ROOM FROM THE OUTSIDE OF THE BUILDING.                           LROOM 49
59 *****LROOM                                                     50
60     R = NROOMS + D                                                    LROOM 51
61     C = RNUM                                                           LROOM 52
62     ROOM(R,C) = TS + ROOM(R,C)                                        LROOM 53
63 *****LROOM                                                     54
64 * INSERT TRANSMISSION COEFFICIENT INTO 'ROOM' ARRAY FOR ENERGY LEAVING LROOM 55

```

65 * A ROOM TO THE OUTSIDE OF THE BUILDING.	LROOM	56
66 *****	LROOM	57
67 R = RNUM	LROOM	58
68 C = NROOMS + D	LROOM	59
69 ROOM(R,C) = TS2 / RAREA(RNUM) + ROOM(R,C)	LROOM	60
70 *****	LROOM	61
71 *	LROOM	62
72 *****	LROOM	63
73 ELSE	LROOM	64
74 *****	LROOM	65
75 * INSERT TRANSMISSION COEFFICIENTS INTO 'ROOM' ARRAY FOR ENERGY GOIN	LROOM	66
76 * FROM ROOM TO ROOM.	LROOM	67
77 *****	LROOM	68
78 R = VAL (FROM(1:2))	LROOM	69
79 C = VAL (TO(1:2))	LROOM	70
80 ROOM(R,C) = TS2 / RAREA(R) + ROOM(R,C)	LROOM	71
81 ROOM(C,R) = TS2 / RAREA(C) + ROOM(C,R)	LROOM	72
82 END IF	LROOM	73
83 RETURN	LROOM	74
84 END	LROOM	75

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	222B	INTEGER	
D	225B	INTEGER	
FROM	3 DUMMY-ARG	CHAR*3	
NROOMS	1244B /ROOMN/	INTEGER	
R	223B	INTEGER	
RAREA	1245B /ROOMN/	REAL	20
RNUM	224B	INTEGER	
ROOM	0B /ROOMN/	REAL	676
TO	4 DUMMY-ARG	CHAR*3	
TS	1 DUMMY-ARG	REAL	
TS2	2 DUMMY-ARG	REAL	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

RMAX	INTEGER	20
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

VAL	INTEGER	1	FUNCTION
-----	---------	---	----------

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

LROOM	5B	4
-------	----	---

FTN 5 1+S52 84/03/14. 10.18.23 PAGE 28
SUBROUTINE LROOM 74/175 OPT=0

--STATISTICS--

PROGRAM-UNIT LENGTH	230B = 152
CM LABELLED COMMON LENGTH	1271B = 697
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.100 SECONDS

1	INTEGER FUNCTION VAL(STRING)	LROOM	76
2	*[[[LROOM	77
3	*[[[[[[LROOM	78
4	*[[[RETURNS THE INTEGER VALUE OF A STRING.	[[[LROOM	79
5	*[[[[[[LROOM	80
6	*[[[[[[LROOM	81
7	*****	*****LROOM	82
8	INTEGER NUMBER, X, L, EXP, DIGIT, GETLEN	LROOM	83
9	CHARACTER * (*) STRING	LROOM	84
10	*	LROOM	85
11	L = GETLEN(STRING)	LROOM	86
12	NUMBER = 0	LROOM	87
13	DO 10 X = L,1,-1	LROOM	88
14	EXP = L - X	LROOM	89
15	DIGIT = ICHAR(STRING(X:X)) - 16	LROOM	90
16	NUMBER = NUMBER + DIGIT*10**EXP	LROOM	91
17	10 CONTINUE	LROOM	92
18	VAL = NUMBER	LROOM	93
19	RETURN	LROOM	94
20	END	LROOM	95

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

DIGIT	76B	INTEGER
EXP	75B	INTEGER
L	74B	INTEGER
NUMBER	72B	INTEGER
STRING	1 DUMMY-ARG	CHAR*(*)
VAL	71B	INTEGER
X	73B	INTEGER

--PROCEDURES-- (LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS-- (LO=A)

- LABEL-ADDRESS-----PROPERTIES----DEF

10 INACTIVE DO-TERM 17

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS--

VAL 6B 1

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.039 SECONDS

1	INTEGER FUNCTION GETLEN (STRING)	LROOM	96
2	*[[[LROOM	97
3	*[[[[[[LROOM	98
4	*[[[DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING	[[[LROOM	99
5	*[[[[[[LROOM	100
6	*[[[[[[LROOM	101
7	*****LROOM	LROOM	102
8	*	LROOM	103
9	* ARGUMENT DEFINITIONS --	LROOM	104
10	* INPUT ARGUMENTS	LROOM	105
11	* STRING - STRING WHOSE LENGTH IS TO BE DETERMINED	LROOM	106
12	*	LROOM	107
13	*****	LROOM	108
14	CHARACTER * (*) STRING	LROOM	109
15	*****	LROOM	110
16	* FUNCTION PARAMETERS	LROOM	111
17	*****	LROOM	112
18	CHARACTER * 1 BLANK	LROOM	113
19	PARAMETER (BLANK = ' ')	LROOM	114
20	*****	LROOM	115
21	* LOCAL VARIABLES	LROOM	116
22	*****	LROOM	117
23	INTEGER NEXT	LROOM	118
24	*****	LROOM	119
25	* START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK	LROOM	120
26	*****	LROOM	121
27	DO 10 NEXT = LEN(STRING),1,-1	LROOM	122
28	IF (STRING(NEXT : NEXT) .NE. BLANK) THEN	LROOM	123
29	GETLEN = NEXT	LROOM	124
30	RETURN	LROOM	125
31	END IF	LROOM	126
32	10 CONTINUE	LROOM	127
33	*****	LROOM	128
34	* ALL CHARACTERS ARE BLANKS	LROOM	129
35	*****	LROOM	130
36	GETLEN = 0	LROOM	131
37	*	LROOM	132
38	RETURN	LROOM	133
39	END	LROOM	134

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE
```

```

GETLEN      63B      INTEGER
NEXT        64B      INTEGER
STRING      1      DUMMY-ARG  CHAR*(*)

```

```
--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE-----VALUE
```

BLANK CHAR*1

```
--PROCEDURES--(LO=A)
-NAME-----TYPE-----ARGS-----CLASS-----
```

LEN	INTEGER	1	INTRINSIC
-----	---------	---	-----------

FTN 5.1+552 84/03/14. 10.18.23 PAGE 31
FUNCTION GETLEN 74/175 OPT=0

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

10 INACTIVE DO-TERM 32

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

GETLEN 6B 1

--STATISTICS--

PROGRAM-UNIT LENGTH	70B = 56
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.039 SECONDS

[illegible]

65	*****	COMT	30
66	*****	LTDB	25
67	* DECLARATION OF VARIABLES	LTDB	26
68	*****	LTDB	27
69	INTEGER NEXT, LAST, R	LTDB	28
70	REAL LATTEN, OATTEN, MATTEN, ATTEN	LTDB	29
71	REAL HEIGHT, WIDTH, AREA	LTDB	30
72	CHARACTER * 3 MAT, ID	LTDB	31
73	LOGICAL NEWTYPE, TYPEEND	LTDB	32
74	*****	LTDB	33
75	*	LTDB	34
76	*****	LTDB	35
77	TDBTOT = 0	LTDB	36
78	DO 10 R = 1, TTOT	LTDB	37
79	*****	LTDB	38
80	* SET TYPEEND CONDITION	LTDB	39
81	*****	LTDB	40
82	NEXT = R + 1	LTDB	41
83	IF (R .EQ. TTOT) THEN	LTDB	42
84	TYPEEND = .TRUE.	LTDB	43
85	ELSE IF (TYPE(R,1) .NE. TYPE(NEXT,1)) THEN	LTDB	44
86	TYPEEND = .TRUE.	LTDB	45
87	ELSE	LTDB	46
88	TYPEEND = .FALSE.	LTDB	47
89	END IF	LTDB	48
90	*****	LTDB	49
91	* SET NEWTYPE CONDITION	LTDB	50
92	*****	LTDB	51
93	LAST = R - 1	LTDB	52
94	IF (R .EQ. 1) THEN	LTDB	53
95	NEWTYPE = .TRUE.	LTDB	54
96	ELSE IF (TYPE(R,1) .NE. TYPE(LAST,1)) THEN	LTDB	55
97	NEWTYPE = .TRUE.	LTDB	56
98	ELSE	LTDB	57
99	NEWTYPE = .FALSE.	LTDB	58
100	END IF	LTDB	59
101	*****	LTDB	60
102	* CALCULATE	LTDB	61
103	*****	LTDB	62
104	IF (NEWTYPE) THEN	LTDB	63
105	*****	LTDB	64
106	* ..INITIALIZE TYPE CONDITIONS	LTDB	65
107	*****	LTDB	66
108	OATTEN = 0	LTDB	67
109	END IF	LTDB	68
110	*****	LTDB	69
111	* ...CALCULATE ATTENUATION FACTOR OF LAYER	LTDB	70
112	*****	LTDB	71
113	MAT = TYPE(R,2)	LTDB	72
114	MATTEN = ATTEN (MAT,FREQ,AFLAG)	LTDB	73
115	T = TDIM(R,3)	LTDB	74
116	LATTEN = MATTEN * T	LTDB	75
117	*****	LTDB	76
118	* ...CALCULATE RUNNING ATTENUATION FACTOR OF OPENING	LTDB	77
119	*****	LTDB	78
120	OATTEN = OATTEN + LATTEN	LTDB	79
121	IF (TYPEEND) THEN	LTDB	80
122	*****	LTDB	81
123	* ...CALCULATE TOTAL OPENING AREA	LTDB	82
124	*****	LTDB	83
125	HEIGHT = TDIM(R,1)	LTDB	84
126	WIDTH = TDIM(R,2)	LTDB	85
127	AREA = HEIGHT * WIDTH	LTDB	86
128	*****	LTDB	87

129 *	INSERT ID, ATTENUATION AND AREA INTO TYPE DATABASE ARRAYS	LTDB	88
130	*****	LTDB	89
131	TDBTOT = TDBTOT + 1	LTDB	90
132	ID = TYPE(R,1)	LTDB	91
133	TDB1(TDBTOT) = ID	LTDB	92
134	TDB2(TDBTOT,1) = OATTEN	LTDB	93
135	TDB2(TDBTOT,2) = AREA	LTDB	94
136	END IF	LTDB	95
137	10 CONTINUE	LTDB	96
138	RETURN	LTDB	97
139	END	LTDB	98

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
AREA	221B		REAL	
BLDG	0B	/INITILC/	CHAR*5	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
HEIGHT	217B		REAL	
ID	223B		CHAR*3	
LAST	212B		INTEGER	
LATTEN	214B		REAL	
MAT	222B		CHAR*3	
MATTEN	216B		REAL	
NEWTYP	224B		LOGICAL	
NEXT	211B		INTEGER	
OATTEN	215B		REAL	
QUALITY	1B	/INITILN/	INTEGER	
R	213B		INTEGER	
RFLAG	3B	/INITILN/	REAL	
T	227B		REAL	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPECL/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPECL/	CHAR*3	105
TYPEEND	225B		LOGICAL	
WIDTH	220B		REAL	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

ATTEN	REAL	3	FUNCTION
-------	------	---	----------

FTN 5.1+552 84/03/14. 10.18.23 PAGE 35
SUBROUTINE LTDB 74/175 OPT=0

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

10 INACTIVE DO-TERM 137

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

LTDB 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH	232B = 154
CM LABELLED COMMON LENGTH	470B = 312
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.116 SECONDS

```

1 SUBROUTINE LTYPE
2 *[[[
3 *[[[
4 *[[[ LOAD THE "TYPE" ARRAYS FROM THE TYPE DATA FILE
5 *[[[
6 *[[[
7 *****LTYPE
8 *****COMT
9 *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT
10 *****COMT
11 INTEGER TMAX COMT
12 PARAMETER (TMAX=35) COMT
13 COMMON /TYDEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR COMT
14 COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX) COMT
15 INTEGER TTOT,TDBTOT,TERR COMT
16 REAL TDIM,TDB2 COMT
17 CHARACTER * 3 TYPE,TDB1 COMT
18 *=====
19 * DESCRIPTION OF ARRAYS COMT
20 *=====
21 * ID MATERIAL FRAME MATERIAL COMT
22 *----- COMT
23 *TYPE(X,1) TYPE(X,2) TYPE(X,3) COMT
24 * A3 A3 A3 COMT
25 *----- COMT
26 * HEIGHT WIDTH LAYER DISTANCE COMT
27 * THICKNESS ABOVE FLOOR COMT
28 *----- COMT
29 * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4) COMT
30 * F8.2 F8.2 F8.2 F8.2 COMT
31 *----- COMT
32 * ID ATTENUATION AREA COMT
33 * ----- COMT
34 * TDB1(X) TDB2(X,1) TDB2(X,2) COMT
35 * A3 E9.3 E9.3 COMT
36 *****COMT
37 *****COMT
38 *****COMF
39 *** COMMON FOR INITIAL PARAMETERS ***COMF
40 *****COMF
41 INTEGER FMAX COMF
42 PARAMETER (FMAX = 50) COMF
43 COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF
44 $ FTOT COMF
45 COMMON /INITILC/ BLDG COMF
46 CHARACTER * 5 BLDG COMF
47 REAL FREQ, AFLAG, RFLAG, FREQA COMF
48 INTEGER QUALITY, FERR, FTOT COMF
49 *****COMF
50 *****COMF
51 *****LTYPE
52 * DECLARATION OF VARIABLES LTYPE
53 ***** LTYPE
54 INTEGER GETLEN, R, C LTYPE
55 CHARACTER * 7 PFN LTYPE
56 ***** LTYPE
57 * LTYPE
58 ***** LTYPE
59 PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'T' LTYPE
60 TERR = 0 LTYPE
61 CALL PF ('GET',0,PFN(1:ETLEN(PFN)), 'RC',TERR) LTYPE
62 IF (TERR .EQ. 0 ) THEN LTYPE
63 OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED', LTYPE
64 $ STATUS='OLD', ACCESS='SEQUENTIAL') LTYPE

```

65	1000	FORMAT (1X,3(1X,A3),4(1X,F8.2))	LTYPE	24
66		TTOT = 0	LTYPE	25
67		DO 10 R = 1,TMAX	LTYPE	26
68		READ (3,1000,END=20)(TYPE(R,C),C=1,3),(TDIM(R,C),C=1,4)	LTYPE	27
69		TTOT = TTOT + 1	LTYPE	28
70	10	CONTINUE	LTYPE	29
71	20	CONTINUE	LTYPE	30
72		CLOSE(3,STATUS='DELETE')	LTYPE	31
73		ELSE IF (TERR .EQ. 2) THEN	LTYPE	32
74		CALL WARNING (5)	LTYPE	33
75		ELSE	LTYPE	34
76		CALL WARNING (6)	LTYPE	35
77		END IF	LTYPE	36
78		RETURN	LTYPE	37
79		END	LTYPE	38

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	236B		INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
PFN	237B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
R	235B		INTEGER	
RFLAG	3B	/INITILN/	REAL	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPEPC/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPEPC/	CHAR*3	105

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	70
20	111B		71
1000	147B	FORMAT	65

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

LTYPE 5B 0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE3 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	245B = 165
CM LABELLED COMMON LENGTH	470B = 312
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.103 SECONDS

```

1 SUBROUTINE LWALL
2 *[[[
3 *[[[
4 *[[[ LOAD THE CONTENTS OF THE FILE 'WALLS' INTO ARRAYS WALL AND WDIM. LWALL
5 *[[[
6 *[[[
7 *****LWALL
8 *****COMW
9 *** COMMON FOR DATABASE OF WALL PARAMETERS ***COMW
10 *****COMW
11 INTEGER WMAX COMW
12 PARAMETER (WMAX = 75) COMW
13 COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR COMW
14 COMMON /WALLC/ WALL(WMAX,4) COMW
15 INTEGER WTOT,WERR COMW
16 REAL WDIM COMW
17 CHARACTER *3 WALL COMW
18 * ===== COMW
19 ** DESCRIPTION OF ARRAYS COMW
20 * ===== COMW
21 * WALL IDENTIFICATION COMW
22 * ----- COMW
23 * DIRECTION FROM TO COMW
24 * ROOM ROOM COMW
25 * ----- COMW
26 * WALL(X,1) WALL(X,2) WALL(X,3) COMW
27 * A3 A3 A3 COMW
28 * ===== COMW
29 * WALL PARAMETERS COMW
30 * ----- COMW
31 * MATERIAL HEIGHT WIDTH LAYER THICKNESS COMW
32 * ----- COMW
33 * WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3) COMW
34 * A3 F8.2 F8.2 F8.2 COMW
35 *****COMW
36 *****COMW
37 *****COMF
38 *** COMMON FOR INITIAL PARAMETERS ***COMF
39 *****COMF
40 INTEGER FMAX COMF
41 PARAMETER (FMAX = 50) COMF
42 COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF
43 $ FTOT COMF
44 COMMON /INITILC/ BLDG COMF
45 CHARACTER * 5 BLDG COMF
46 REAL FREQ, AFLAG, RFLAG, FREQA COMF
47 INTEGER QUALITY, FERR, FTOT COMF
48 *****COMF
49 *****COMF
50 ***** LWALL
51 * DECLARATION OF VARIABLES LWALL
52 ***** LWALL
53 INTEGER GETLEN, R, C LWALL
54 CHARACTER * 7 NAME, PFN LWALL
55 ***** LWALL
56 * LWALL
57 ***** LWALL
58 NAME = 'B'//BLDG(1:GETLEN(BLDG))//'W' LWALL
59 PFN = NAME (1:GETLEN(NAME)) LWALL
60 WERR = 0 LWALL
61 CALL PF ('GET',0,PFN(1:GETLEN(PFN)), 'RC',WERR) LWALL
62 IF ( WERR .EQ. 0 ) THEN LWALL
63 OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED', LWALL
64 $ STATUS='OLD', ACCESS='SEQUENTIAL') LWALL

```


65	1000	FORMAT (1X,4(1X,A3),3(1X,F8.2))	LWALL	25
66		WTOT = 0	LWALL	26
67		DO 10 R = 1,WMAX	LWALL	27
68		READ (3,1000,END=20)(WALL(R,C),C=1,4),(WDIM(R,C),C=1,3)	LWALL	28
69		WTOT = WTOT + 1	LWALL	29
70	10	CONTINUE	LWALL	30
71	20	CONTINUE	LWALL	31
72		CLOSE(3,STATUS='DELETE')	LWALL	32
73		ELSE IF (WERR .EQ. 2) THEN	LWALL	33
74		CALL WARNING (7)	LWALL	34
75		ELSE	LWALL	35
76		CALL WARNING (8)	LWALL	36
77		END IF	LWALL	37
78		RETURN	LWALL	38
79		END	LWALL	39

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	255B		INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
NAME	256B		CHAR*7	
PFN	257B		CHAR*7	
QUALITY	1B	/INITILN/	INTEGER	
R	254B		INTEGER	
RFLAG	3B	/INITILN/	REAL	
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

FMAX	INTEGER	50
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	70
20	117B		71
1000	155B	FORMAT	65

FTN 5.1+552 84/03/14. 10.18.23 PAGE 41
SUBROUTINE LWALL 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

LWALL 5B 0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE3 AUX/FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	265B = 181
CM LABELLED COMMON LENGTH	566B = 374
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.108 SECONDS

1	SUBROUTINE PHOLE		PHOLE	1
2	*[[[[[[PHOLE		2
3	*[[[[[[PHOLE		3
4	*[[[PRINT OUT THE CONTENTS OF THE HOLE ARRAY	[[[PHOLE		4
5	*[[[[[[PHOLE		5
6	*[[[[[[PHOLE		6
7	*****PHOLE			7
8	*****COMH			1
9	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS ***	COMH		2
10	*****COMH			3
11	INTEGER HMAX	COMH		4
12	PARAMETER (HMAX = 35)	COMH		5
13	COMMON /HOLEN/ HTOT, HERR	COMH		6
14	COMMON /HOLEC/ HOLE(HMAX,4)	COMH		7
15	INTEGER HTOT, HERR	COMH		8
16	CHARACTER * 3 HOLE	COMH		9
17	* =====	COMH		10
18	* DESCRIPTION OF ARRAYS	COMH		11
19	* =====	COMH		12
20	ROOM IDENTIFICATION APERTURE ID	COMH		13
21	* -----	COMH		14
22	* DIRECTION FROM ROOM TO ROOM	COMH		15
23	* -----	COMH		16
24	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH		17
25	* A3 A3 A3 A3	COMH		18
26	*****COMH			19
27	*****COMH			20
28	INTEGER R,C	PHOLE		9
29	PRINT *	PHOLE		10
30	PRINT *, ' DOOR AND WINDOW LOCATIONS'	PHOLE		11
31	PRINT*, '*****'	PHOLE		12
32	PRINT*, ' WALL IDENTIFICATION'	PHOLE		13
33	PRINT*, ' ----- '	PHOLE		14
34	PRINT*, ' ID DIRECTION FROM TO'	PHOLE		15
35	PRINT*, '==== ===== '	PHOLE		16
36	DO 10 R = 1,HTOT	PHOLE		17
37	PRINT 1000,HOLE(R,4),(HOLE(R,C),C=1,3)	PHOLE		18
38	10 CONTINUE	PHOLE		19
39	PRINT*, '===== '	PHOLE		20
40	1000 FORMAT (2X,A3,5X,3(3X,A3))	PHOLE		21
41	RETURN	PHOLE		22
42	END	PHOLE		23

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	174B		INTEGER	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
R	173B		INTEGER	

-NAME----	TYPE-----	VALUE-----
-----------	-----------	------------

HMAX INTEGER 35

FTN 5.1+552 84/03/14. 10.18.23 PAGE 43
SUBROUTINE PHOLE 74/175 OPT=0

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	38
1000	124B	FORMAT	40

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

PHOLE	5B	0
-------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	201B = 129
CM LABELLED COMMON LENGTH	54B = 44
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.062 SECONDS

-VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

251

FTN 5.1+552 84/03/14. 10.18.23 PAGE 45
SUBROUTINE PROOM 74/175 OPT=0

RAREA	1245E	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
RMAX	INTEGER	20

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	41
20	INACTIVE	DO-TERM	44
1000	154B	FORMAT	47

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

PROOM	5B	0
-------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	217B =	143
CM LABELLED COMMON LENGTH	1362B =	754
CM STORAGE USED	61000B =	25088
COMPILE TIME	0.080	SECONDS

```

1      SUBROUTINE PTDB                                PTDB      1
2      *****PTDB                                PTDB      2
3      * PRINT SUMMARY OF THE ATTENUATION OF THE DOORS AND WINDOWS PTDB      3
4      *****PTDB                                PTDB      4
5      *****COMT                                COMT      1
6      *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT      2
7      *****COMT                                COMT      3
8      INTEGER TMAX                                COMT      4
9      PARAMETER (TMAX=35)                        COMT      5
10     COMMON /TYPEN/ TDIM(TMAX,4), TTOT, TDB2(TMAX,2), TDBTOT, TERR COMT      6
11     COMMON /YPEC/ TYPE(TMAX,3), TDB1(TMAX)      COMT      7
12     INTEGER TTOT, TDBTOT, TERR                  COMT      8
13     REAL TDIM, TDB2                             COMT      9
14     CHARACTER * 3 TYPE, TDB1                   COMT     10
15     *=====COMT                                COMT     11
16     * DESCRIPTION OF ARRAYS                      COMT     12
17     *=====COMT                                COMT     13
18     * ID MATERIAL FRAME MATERIAL                COMT     14
19     *-----COMT                                COMT     15
20     *TYPE(X,1) TYPE(X,2) TYPE(X,3)              COMT     16
21     * A3 A3 A3                                  COMT     17
22     *=====COMT                                COMT     18
23     * HEIGHT WIDTH LAYER DISTANCE              COMT     19
24     * THICKNESS ABOVE FLOOR                    COMT     20
25     *-----COMT                                COMT     21
26     * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4) COMT     22
27     * F8.2 F8.2 F8.2 F8.2                     COMT     23
28     *=====COMT                                COMT     24
29     * ID ATTENUATION AREA                      COMT     25
30     *-----COMT                                COMT     26
31     * TDB1(X) TDB2(X,1) TDB2(X,2)              COMT     27
32     * A3 E9.3 E9.3                             COMT     28
33     *****COMT                                COMT     29
34     *****COMT                                COMT     30
35     INTEGER R,C                                PTDB      6
36     CHARACTER * 3 ID                          PTDB      7
37     PRINT *                                    PTDB      8
38     PRINT*, 'DOOR AND WINDOW SUMMARY'          PTDB      9
39     PRINT*, '*****'                          PTDB     10
40     PRINT*, 'ID ATTENUATION AREA'              PTDB     11
41     PRINT*, '===== '                        PTDB     12
42     DO 10 R = 1, TDBTOT                        PTDB     13
43     PRINT 1000, TDB1(R), (TDB2(R,C), C=1,2)    PTDB     14
44 10 CONTINUE                                  PTDB     15
45     PRINT*, '===== '                        PTDB     16
46 1000 FORMAT (1X,A3,5X,F8.2,5X,F6.2)           PTDB     17
47     RETURN                                    PTDB     18
48     END                                        PTDB     19

```

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	144B		INTEGER	
ID	NONE	UNUSED/*S*	CHAR*3	
R	143B		INTEGER	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/YPEC/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/YPEC/	CHAR*3	105

FTN 5 1+552 84/03/14. 10.18.23 PAGE 47
SUBROUTINE PTDB 74/175 OPT=0

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

TMAX INTEGER 35

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	44
1000	107E	FORMAT	46

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

PTDB 5E 0

--STATISTICS--

PROGRAM-UNIT LENGTH	151B = 105
CM LABELLED COMMON LENGTH	377B = 255
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.060 SECONDS

741175 OPT=0

[illegible]

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	217B	INTEGER	
R	216B	INTEGER	
TDETOT	323B /TYPE/	INTEGER	
TDB1	37B /TYPE/	CHAR*3	35
TDB2	215B /TYPE/	REAL	70
TDIM	0B /TYPE/	REAL	140

FTN 5.1+552 84/03/14. 10.18.23 PAGE 49
SUBROUTINE PTYPE 74/175 OPT=0

TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPEPC/	CHAR*3	105

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

TMAX	INTEGER	35
------	---------	----

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	49
1000	152B	FORMAT	51

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

PTYPE	5B	0
-------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	225B = 149
CM LABELLED COMMON LENGTH	377B = 255
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.069 SECONDS

1	SUBROUTINE PWALL	PWALL	1
2	*****	PWALL	2
3	*** PRINT OUT THE CONTENTS OF THE WALL AND WDIM ARRAYS	PWALL	3
4	*****	PWALL	4
5	*	PWALL	5
6	*****	COMW	1
7	*** COMMON FOR DATABASE OF WALL PARAMETERS	***COMW	2
8	*****	COMW	3
9	INTEGER WMAX	COMW	4
10	PARAMETER (WMAX = 75)	COMW	5
11	COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR	COMW	6
12	COMMON /WALLC/ WALL(WMAX,4)	COMW	7
13	INTEGER WTOT,WERR	COMW	8
14	REAL WDIM	COMW	9
15	CHARACTER *3 WALL	COMW	10
16	* =====	COMW	11
17	** DESCRIPTION OF ARRAYS	COMW	12
18	* =====	COMW	13
19	* WALL IDENTIFICATION	COMW	14
20	* -----	COMW	15
21	* DIRECTION FROM TO	COMW	16
22	* ROOM ROOM	COMW	17
23	* -----	COMW	18
24	* WALL(X,1) WALL(X,2) WALL(X,3)	COMW	19
25	* A3 A3 A3	COMW	20
26	* =====	COMW	21
27	* WALL PARAMETERS	COMW	22
28	* -----	COMW	23
29	* MATERIAL HEIGHT WIDTH LAYER THICKNESS	COMW	24
30	* -----	COMW	25
31	* WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3)	COMW	26
32	* A3 F8.2 F8.2 F8.2	COMW	27
33	*****	COMW	28
34	*****	COMW	29
35	*	PWALL	7
36	INTEGER R, C	PWALL	8
37	*	PWALL	9
38	PRINT*	PWALL	10
39	PRINT *, 'WALL IDENTIFICATION WALL PARAMETERS'	PWALL	11
40	PRINT *, '*****'	PWALL	12
41	PRINT *, ' DIR FROM TO MATERIAL HEIGHT WIDTH THICKNESS'	PWALL	13
42	PRINT *, '====='	PWALL	14
43	DO 10 R = 1,WTOT	PWALL	15
44	PRINT 1000, (WALL(R,C),C=1,4), (WDIM(R,C),C=1,3)	PWALL	16
45	10 CONTINUE	PWALL	17
46	PRINT *, '====='	PWALL	18
47	1000 FORMAT (1X,3(2X,A3),10X,A3,1X,3(1X,F7.2))	PWALL	19
48	RETURN	PWALL	20
49	END	PWALL	21

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	205B	INTEGER	
R	204B	INTEGER	
WALL	0B /WALLC/	CHAR*3	300
WDIM	0B /WALLN/	REAL	225
WERR	342B /WALLN/	INTEGER	
WTOT	341B /WALLN/	INTEGER	

FTN 5.1+552 84/03/14. 10.18.23 PAGE 51
SUBROUTINE PWALL 74/175 OPT=0

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

WMAX INTEGER 75

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	45
1000	142B	FORMAT	47

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

PWALL 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH	213B = 139
CM LABELLED COMMON LENGTH	475B = 317
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.063 SECONDS

259

65	EXACT = .TRUE.	RCOEF	52
66	CINDEX = C	RCOEF	53
67	RCOEF = MRCOEF (RINDEX,CINDEX)	RCOEF	54
68	RCOEF = RCOEF * (1 + RFLAG / 100)	RCOEF	55
69	END IF	RCOEF	56
70 20	CONTINUE	RCOEF	57
71	*****	RCOEF	58
72 *	INTERPOLATE REFLECTION COEFFICIENT VALUES IF EXACT FREQUENCY IS	RCOEF	59
73 *	NOT IN THE FREQUENCY/REFLECTION COEFFICIENT ARRAYS.	RCOEF	60
74	*****	RCOEF	61
75	IF (.NOT. EXACT) THEN	RCOEF	62
76	DO 30 C=1,6	RCOEF	63
77	IF (FREQ .GT. MFREQ (RINDEX,C) .AND.	RCOEF	64
78	5 FREQ .LT. MFREQ (RINDEX,C+1)) THEN	RCOEF	65
79	CINDEX = C	RCOEF	66
80	END IF	RCOEF	67
81 30	CONTINUE	RCOEF	68
82	F = ALOG10 (FREQ)	RCOEF	69
83	LOFREQ = ALOG10 (MFREQ (RINDEX, CINDEX))	RCOEF	70
84	HIFREQ = ALOG10 (MFREQ (RINDEX, CINDEX + 1))	RCOEF	71
85	LORCOEF = MRCOEF (RINDEX,CINDEX)	RCOEF	72
86	HIRCOEF = MRCOEF (RINDEX, CINDEX + 1)	RCOEF	73
87	FRAC = (F - LOFREQ) / (HIFREQ - LOFREQ)	RCOEF	74
88	RCOEF = LORCOEF + (FRAC * (HIRCOEF - LORCOEF))	RCOEF	75
89	RCOEF = RCOEF * (1 + RFLAG / 100)	RCOEF	76
90	END IF	RCOEF	77
91	RETURN	RCOEF	78
92	END	RCOEF	79

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

C	254B		INTEGER	
CINDEX	256B		INTEGER	
EXACT	260B		LOGICAL	
F	252B		REAL	
FOUND	257B		LOGICAL	
FRAC	243B		REAL	
FREQ	2	DUMMY-ARG	REAL	
HIFREQ	247B		REAL	
HIRCOEF	251B		REAL	
IERR	262B		INTEGER	
LOFREQ	246B		REAL	
LORCOEF	250B		REAL	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	1	DUMMY-ARG	CHAR*3	
MATTEN	0B	/MATN/	REAL	700
MAXFREQ	245B		REAL	
MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MINFREQ	244B		REAL	
MRCOEF	1274B	/MATN/	REAL	700
MTOT	4375B	/MATN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
R	253B		INTEGER	
RCOEF	242B		REAL	
RFLAG	3	DUMMY-ARG	REAL	
RINDEX	255B		INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

MMA	INTEGER		100
-----	---------	--	-----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

ALOG10	REAL	1	INTRINSIC
ERROR		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	44
20	INACTIVE	DO-TERM	70
30	INACTIVE	DO-TERM	81

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

RCOEF	6B	3
-------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	267B = 183
CM LABELLED COMMON LENGTH	5730B = 3032
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.127 SECONDS


```

1 SUBROUTINE SRCHTDB (ID, OATTEN,OAREA) SRCHTDB 1
2 *[[[[]]]]SRCHTDB 2
3 *[[[[]]]]SRCHTDB 3
4 *[[[ GIVEN AN OPENING ID, THIS SUBROUTINE RETURNS ITS ATTENUATION ]]]SRCHTDB 4
5 *[[[ AND AREA. ]]]SRCHTDB 5
6 *[[[[]]]]SRCHTDB 6
7 *[[[[]]]]SRCHTDE 7
8 *****COMT 1
9 *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT 2
10 *****COMT 3
11 INTEGER TMAX COMT 4
12 PARAMETER (TMAX=35) COMT 5
13 COMMON /TYPEIN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR COMT 6
14 COMMON /YPEEC/TYPE(TMAX,3),TDB1(TMAX) COMT 7
15 INTEGER TTOT,TDBTOT,TERR COMT 8
16 REAL TDIM,TDB2 COMT 9
17 CHARACTER * 3 TYPE,TDB1 COMT 10
18 *=====COMT 11
19 * DESCRIPTION OF ARRAYS COMT 12
20 *=====COMT 13
21 * ID MATERIAL FRAME MATERIAL COMT 14
22 *-----COMT 15
23 *TYPE(X,1) TYPE(X,2) TYPE(X,3) COMT 16
24 * A3 A3 A3 COMT 17
25 *=====COMT 18
26 * HEIGHT WIDTH LAYER DISTANCE COMT 19
27 * THICKNESS ABOVE FLOOR COMT 20
28 *-----COMT 21
29 * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4) COMT 22
30 * F8.2 F8.2 F8.2 F8.2 COMT 23
31 *=====COMT 24
32 * ID ATTENUATION AREA COMT 25
33 * -----COMT 26
34 * TDB1(X) TDB2(X,1) TDB2(X,2) COMT 27
35 * A3 E9.3 E9.3 COMT 28
36 *****COMT 29
37 *****COMT 30
38 INTEGER R SRCHTDB 9
39 REAL OAREA,OATTEN SRCHTDB 10
40 CHARACTER *3 ID SRCHTDB 11
41 * SRCHTDB 12
42 DO 10 R = 1 , TDBTOT SRCHTDB 13
43 IF (TDB1(R).EQ.ID) THEN SRCHTDB 14
44 OATTEN = TDB2(R,1) SRCHTDB 15
45 OAREA = TDB2(R,2) SRCHTDB 16
46 RETURN SRCHTDB 17
47 END IF SRCHTDB 18
48 10 CONTINUE SRCHTDB 19
49 END SRCHTDB 20

```

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ID	1	DUMMY-ARG	CHAR*3	
OAREA	3	DUMMY-ARG	REAL	
OATTEN	2	DUMMY-ARG	REAL	
R	65B		INTEGER	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPECL/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0E	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TTOT	214B	/TYPEN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)
-NAME---TYPE-----VALUE

TMAX INTEGER 35

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

10 INACTIVE DO-TERM 48

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

SRCHTDB 5B 3

--STATISTICS--

PROGRAM-UNIT LENGTH 71B = 57
CM LABELLED COMMON LENGTH 377B = 255
CM STORAGE USED 61000B = 25088
COMPILE TIME 0.049 SECONDS

1	SUBROUTINE WARNING(ERR)	WARNING	1
2	INTEGER ERR, ERRM	WARNING	2
3	CHARACTER*45 MESSAGE(20)	WARNING	3
4	DATA MESSAGE(1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	4
5	DATA MESSAGE(2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE' /	WARNING	5
6	DATA MESSAGE(3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	6
7	DATA MESSAGE(4)/'FILE HANDLING PROBLEM ON "MATTER" FILE' /	WARNING	7
8	DATA MESSAGE(5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	8
9	DATA MESSAGE(6)/'FILE HANDLING PROBLEM ON "TYPE" FILE' /	WARNING	9
10	DATA MESSAGE(7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	10
11	DATA MESSAGE(8)/'FILE HANDLING PROBLEM ON "WALL" FILE' /	WARNING	11
12	DATA MESSAGE(9)/'HEIGHT AND WIDTH OF ROOM MISSING' /	WARNING	12
13	DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING' /	WARNING	13
14	DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG' /	WARNING	14
15	DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE' /	WARNING	15
16	DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE' /	WARNING	16
17	DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE' /	WARNING	17
18	DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE' /	WARNING	18
19	DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE' /	WARNING	19
20	DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE' /	WARNING	20
21	DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE' /	WARNING	21
22	DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE' /	WARNING	22
23	DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE' /	WARNING	23
24	ERRM=12	WARNING	24
25	IERR = ERR	WARNING	25
26	IF(ERR.GT.ERRM) IERR=20	WARNING	26
27	WRITE(6,20)	WARNING	27
28	WRITE(6,10) ERR,MESSAGE(IERR)	WARNING	28
29	WRITE(6,20)	WARNING	29
30 10	FORMAT(' ***WARNING NUMBER = ',I5,' *** ',A45)	WARNING	30
31 20	FORMAT(' ')	WARNING	31
32	RETURN	WARNING	32
33	END	WARNING	33

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

ERR	1	DUMMY-ARG	INTEGER	
ERRM	60B		INTEGER	
IERR	213B		INTEGER	
MESSAGE	61B		CHAR*45	20

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

10	34B	FORMAT	30
20	42B	FORMAT	31

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

WARNING	5B	1
---------	----	---

FTN 5 1+552 84/03/14. 10.18.23 PAGE 58
SUBROUTINE WARNING 74/175 OPT=0

--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-----

TAPE6 FMT/SEQ

--STATISTICS--

PROGRAM-UNIT LENGTH	216B = 142
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.061 SECONDS

```

1      SUBROUTINE SETUP                                SETUP      1
2      *****COMR                                  COMR      1
3      *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
4      *****COMR                                  COMR      3
5      INTEGER RMAX                                    COMR      4
6      PARAMETER (RMAX = 20)                          COMR      5
7      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
8      INTEGER NROOMS                                COMR      7
9      REAL ROOM                                       COMR      8
10     *****COMR                                  COMR      9
11     *****COMR                                  COMR     10
12     *****COMJ                                  COMJ      1
13     *                                              COMJ      2
14     * COMMON FOR EVALUATION OF ROOM MATRIX          COMJ      3
15     *                                              COMJ      4
16     *****COMJ                                  COMJ      5
17     COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME COMJ      6
18     +,SWR(RMAX,6),IDIR                            COMJ      7
19     REAL TMAT ,ENERGY,POWER,SWR                     COMJ      8
20     LOGICAL FTIME                                   COMJ      9
21     *****COMD                                  COMD      1
22     * COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS COMD      2
23     *****COMD                                  COMD      3
24     COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ),DREFL, DREFLW COMD      4
25     REAL DDABS ,DREFL , DREFLW                     COMD      5
26     *****COMD                                  COMD      6
27     *****COMD                                  COMD      7
28     DATA FTIME /.TRUE./                            SETUP      5
29     IF (FTIME.EQV..FALSE.) GOTO 500                 SETUP      6
30     C*****SETUP                                  SETUP      7
31     C* CALCULATE DIAGONAL ELEMENTS                  SETUP      8
32     C* ASSUME DIAGONAL ELEMENTS ARE INITIALLY ZERO  SETUP      9
33     C*****SETUP                                  SETUP     10
34     DO 200 IR=1,NROOMS                             SETUP     11
35     DIAG =0.0                                        SETUP     12
36     DO 100 IC=1,NROOMS + 6                          SETUP     13
37     100 DIAG =DIAG + ROOM(IR,IC) + DDABS(IR,IC)     SETUP     14
38     200 ROOM(IR,IR) = -DIAG                         SETUP     15
39     C*****SETUP                                  SETUP     16
40     C* SET FTIME FALSE                             SETUP     17
41     C*****SETUP                                  SETUP     18
42     FTIME = .FALSE.                                  SETUP     19
43     C*****SETUP                                  SETUP     20
44     C*****SETUP                                  SETUP     21
45     C* NOW LOAD ROOM INTO TMAT                      SETUP     22
46     C* NOTE THAT THE T MATRIX IS REFLECTED ABOUT THE DIAGONAL SETUP     23
47     C* WITH RESPECT TO THE ROOM MATRIX              SETUP     24
48     C*****SETUP                                  SETUP     25
49     500 DO 600 IR = 1, NROOMS                       SETUP     26
50     DO 600 IC = 1, NROOMS                           SETUP     27
51     600 TMAT(IR,IC) = ROOM(IC,IR)                  SETUP     28
52     RETURN                                           SETUP     29
53     END                                              SETUP     30

```

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

DDABS	0B	/ROOMD/	REAL	676
DIAG	114B		REAL	
DREFL	1244B	/ROOMD/	REAL	
DREFLW	1245B	/ROOMD/	REAL	
ENERGY	620B	/MAT/	REAL	20
FTIME	652B	/MAT/	LOGICAL	

IC	115B		INTEGER	
IDIR	1043B	/MAT/	INTEGER	
IR	112B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
POWER	644B	/MAT/	REAL	6
RAREA	1245B	/ROOMN/	REAL	20
ROOM	0B	/ROOMN/	REAL	676
SWR	653B	/MAT/	REAL	120
TMAT	0B	/MAT/	REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

RMAX	INTEGER	20
------	---------	----

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

100	INACTIVE	DO-TERM	37
200	INACTIVE	DO-TERM	38
500	53B		49
600	INACTIVE	DO-TERM	51

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

SETUP	5B	0
-------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	123B = 83
CM LABELLED COMMON LENGTH	3603B = 1923
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.074 SECONDS

1 C**	FUNCTION DETERM	DETERM	1
2 C**		DETERM	2
3 C**	PURPOSE	DETERM	3
4 C**	CALCULATE THE DETERMINANT OF A SQUARE MATRIX	DETERM	4
5 C**		DETERM	5
6 C**	USAGE	DETERM	6
7 C**	DET=DETERM(ARRAY,NORDER)	DETERM	7
8 C**		DETERM	8
9 C**	DESCRIPTION OF PARAMETERS	DETERM	9
10 C**	ARRAY - MATRIX	DETERM	10
11 C**	NORDER - ORDER OF DETERMINANT(DEGREE OF MATRIX)	DETERM	11
12 C**		DETERM	12
13 C**	COMMENTS	DETERM	13
14 C**	THIS SUBROUTINE DESTROYS THE INPUT MATRIX ARRAY	DETERM	14
15 C**	THIS ROUTINE WAS MODIFIED SO THAT THE MAXIMUM	DETERM	15
16 C**	VALUE IN THE TOP ROW IS MOVED OVER TO THE DIAGONAL	DETERM	16
17	FUNCTION DETERM(ARRAY,NORDER)	DETERM	17
18	DIMENSION ARRAY(20,*)	DETERM	18
19 10	DETERM=1.	DETERM	19
20 11	DO 50 K=1,NORDER	DETERM	20
21 C**		DETERM	21
22 C**	INTERCHANGE COLUMNS IF DIAGONAL ELEMENT IS ZERO	DETERM	22
23 C**		DETERM	23
24 21	AMAX=0.0	DETERM	24
25	JMAX=K	DETERM	25
26	DO 25 J=K,NORDER	DETERM	26
27	TMP=ARRAY(K,J)	DETERM	27
28	TMP=ABS(TMP)	DETERM	28
29	IF(TMP.LT.AMAX) GOTO 25	DETERM	29
30	AMAX=TMP	DETERM	30
31	JMAX=J	DETERM	31
32 25	CONTINUE	DETERM	32
33	J=JMAX	DETERM	33
34	IF(J.GT.K) GOTO 31	DETERM	34
35	AATMP=ABS(ARRAY(K,K))	DETERM	35
36	IF (AATMP .GE. 1.0E-05) GOTO 41	DETERM	36
37 30	DETERM =0.	DETERM	37
38	GOTO 60	DETERM	38
39 31	DO 34 I=K,NORDER	DETERM	39
40	SAVE=ARRAY(I,J)	DETERM	40
41	ARRAY(I,J)=ARRAY(I,K)	DETERM	41
42 34	ARRAY(I,K)=SAVE	DETERM	42
43	DETERM=-DETERM	DETERM	43
44 C**		DETERM	44
45 C**	SUBTRACT ROW K FROM LOWER ROWS TO GET DIAGONAL MATRIX	DETERM	45
46 C**		DETERM	46
47 41	DETERM=DETERM*ARRAY(K,K)	DETERM	47
48	IF(DETERM.EQ.0.0) RETURN	DETERM	48
49	IF(K-NORDER) 43,50,50	DETERM	49
50 43	K1=K+1	DETERM	50
51	DO 46 I=K1,NORDER	DETERM	51
52	DO 46 J=K1,NORDER	DETERM	52
53 46	ARRAY(I,J)=ARRAY(I,J)-ARRAY(I,K)*ARRAY(K,J)/ARRAY(K,K)	DETERM	53
54 50	CONTINUE	DETERM	54
55 60	RETURN	DETERM	55
56	END	DETERM	56

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AATMP	222B		REAL	
AMAX	215B		REAL	
ARRAY	1	DUMMY-ARG	REAL	ADJ-ARY
DETERM	212B		REAL	
I	223B		INTEGER	
J	217B		INTEGER	
JMAX	216B		INTEGER	
K	213B		INTEGER	
K1	226B		INTEGER	
NORDER	2	DUMMY-ARG	INTEGER	
SAVE	225B		REAL	
TMP	221B		REAL	

--PROCEDURES--(LO=A)
 -NAME-----TYPE-----ARGS-----CLASS-----

ABS	GENERIC	1	INTRINSIC
-----	---------	---	-----------

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF -LABEL-ADDRESS-----PROPERTIES----DEF

10	*NO REFS*	19	34	INACTIVE	DO-TERM	42
11	*NO REFS*	20	41	124B		47
21	*NO REFS*	24	43	INACTIVE		50
25	44B DO-TERM	32	46	INACTIVE	DO-TERM	53
30	*NO REFS*	37	50	200B	DO-TERM	54
31	71B	39	60	205B		55

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

DETERM	6B	2
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	233B = 155
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.111 SECONDS

```

1      SUBROUTINE ECALC                                ECALC      1
2      DIMENSION PVECTOR(20)                          ECALC      2
3      REAL NUM                                         ECALC      3
4      LOGICAL TLOW                                     ECALC      4
5      *****COMR                                     COMR      1
6      *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
7      *****COMR                                     COMR      3
8      INTEGER RMAX                                     COMR      4
9      PARAMETER (RMAX = 20)                           COMR      5
10     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
11     INTEGER NROOMS                                    COMR      7
12     REAL ROOM                                         COMR      8
13     *****COMR                                     COMR      9
14     *****COMR                                     COMR     10
15     *****COMJ                                     COMJ      1
16     *                                                 COMJ      2
17     * COMMON FOR EVALUATION OF ROOM MATRIX           COMJ      3
18     *                                                 COMJ      4
19     *****COMJ                                     COMJ      5
20     COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),ETIME COMJ      6
21     +,SWR(RMAX,6),IDIR                               COMJ      7
22     REAL TMAT,ENERGY,POWER,SWR                       COMJ      8
23     LOGICAL ETIME                                    COMJ      9
24     *****ECALC                                    ECALC     10
25     * CALCULATE THE ENERGY BALANCE IN THE ROOMS    ECALC     11
26     *****ECALC                                    ECALC     12
27     *                                                 ECALC     13
28     *****ECALC                                    ECALC     14
29     * CALCULATE THE DENOMINATOR TERM                ECALC     15
30     *****ECALC                                    ECALC     16
31     CALL SETUP                                       ECALC     17
32     DENOM=DETERM(TMAT,NROOMS)                       ECALC     18
33     IF(DENOM) 100,50,100                           ECALC     19
34     *****ECALC                                    ECALC     20
35     * ERROR # 4: DENOMINATOR = 0.                   ECALC     21
36     *****ECALC                                    ECALC     22
37     50 IERR = 4                                     ECALC     23
38     CALL ERROR(IERR)                                ECALC     24
39     RETURN                                           ECALC     25
40     *****ECALC                                    ECALC     26
41     * CALCULATE THE INPUT POWER VECTOR TO EACH ROOM ECALC     27
42     *****ECALC                                    ECALC     28
43 100 TLOW=.TRUE.                                     ECALC     29
44     DO 300 ICOL = 1,NROOMS                          ECALC     30
45     SUM = 0.0                                         ECALC     31
46     DO 200 IPWR = 1,6                                ECALC     32
47     IROW = IPWR + NROOMS                            ECALC     33
48 200 SUM = SUM + POWER(IPWR) * ROOM(IROW,ICOL)       ECALC     34
49     PVECTOR(ICOL) = - SUM                           ECALC     35
50 300 IF (SUM.GT.(1.0E-06)) TLOW=.FALSE.              ECALC     36
51     *****ECALC                                    ECALC     37
52     * CHECK IF INPUT POWER IS TOO LOW               ECALC     38
53     *****ECALC                                    ECALC     39
54     IF (TLOW.NEQV..TRUE.) GOTO 350                 ECALC     40
55     *****ECALC                                    ECALC     41
56     * INPUT TOO LOW                                 ECALC     42
57     *****ECALC                                    ECALC     43
58     DO 310 ICOL=1,NROOMS                            ECALC     44
59 310 ENERGY(ICOL)=1.0E-05                          ECALC     45
60     RETURN                                           ECALC     46
61     *****ECALC                                    ECALC     47
62     * SET UP NUMERATORS                             ECALC
63     *****ECALC                                    ECALC
64     * RENEW TMATRIX                                ECALC

```

65 *****	ECALC	48
66 350 DO 500 ICOL = 1,NROOMS	ECALC	49
67 CALL SETUP	ECALC	50
68 *****	ECALC	51
69 * PUT PVECTOR INTO PROPER COLUMN	ECALC	52
70 *****	ECALC	53
71 DO 400 IROW = 1,NROOMS	ECALC	54
72 400 TMAT(IROW,ICOL) = PVECTOR(IROW)	ECALC	55
73 *****	ECALC	56
74 * NOW CALCULATE THE ENERGY FOR THE ROOM REPRESENTED	ECALC	57
75 * BY ICOL	ECALC	58
76 *****	ECALC	59
77 NUM = DETERM(TMAT,NROOMS)	ECALC	60
78 ENERGY(ICOL) = NUM/DENOM	ECALC	61
79 500 CONTINUE	ECALC	62
80 RETURN	ECALC	63
81 END	ECALC	64

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

DENOM	213B		REAL	
ENERGY	620B	/MAT/	REAL	20
ETIME	652B	/MAT/	LOGICAL	
ICOL	215B		INTEGER	
IDIR	1043B	/MAT/	INTEGER	
IERR	214B		INTEGER	
IPWR	220B		INTEGER	
IROW	222B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
NUM	211B		REAL	
POWER	644B	/MAT/	REAL	6
PVECTOR	165B		REAL	20
RAREA	1245B	/ROOMN/	REAL	20
ROOM	0B	/ROOMN/	REAL	676
SUM	217B		REAL	
SWR	653B	/MAT/	REAL	120
TLOW	212B		LOGICAL	
TMAT	0B	/MAT/	REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

RMAX	INTEGER	20
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

DETERM	REAL	2	FUNCTION
ERROR		1	SUBROUTINE
SETUP		0	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF -LABEL-ADDRESS-----PROPERTIES-----DEF

50	INACTIVE	37	310	INACTIVE	DO-TERM	59
100	22B	43	350	107B		66
200	INACTIVE	DO-TERM	48	400	INACTIVE	DO-TERM
300	INACTIVE	DO-TERM	50	500	INACTIVE	DO-TERM

FTN 5.1+552 84/03/14. 10.18.23 PAGE 65
SUBROUTINE ECALC 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS---ARGS---

ECALC 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH	230B = 152
CM LABELLED COMMON LENGTH	2335B = 1245
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.101 SECONDS

```

1      SUBROUTINE PPWR                                PPWR      1
2      *****COMF                                  1
3      *** COMMON FOR INITIAL PARAMETERS                ***COMF      2
4      *****COMF                                  3
5      INTEGER FMAX                                COMF      4
6      PARAMETER (FMAX = 50)                        COMF      5
7      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
8      $      FTOT                                COMF      7
9      COMMON /INITILC/ BLDG                        COMF      8
10     CHARACTER * 5 BLDG                            COMF      9
11     REAL FREQ, AFLAG, RFLAG, FREQA                COMF     10
12     INTEGER QUALITY, FERR, FTOT                    COMF     11
13     *****COMF                                  12
14     *****COMF                                  13
15     *****COMR                                  1
16     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
17     *****COMR                                  3
18     INTEGER RMAX                                COMR      4
19     PARAMETER (RMAX = 20)                        COMR      5
20     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
21     INTEGER NROOMS                                COMR      7
22     REAL ROOM                                    COMR      8
23     *****COMR                                  9
24     *****COMR                                  10
25     *****COMJ                                  1
26     *                                            COMJ      2
27     * COMMON FOR EVALUATION OF ROOM MATRIX        COMJ      3
28     *                                            COMJ      4
29     *****COMJ                                  5
30     COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME COMJ      6
31     +,SWR(RMAX,6),IDIR                            COMJ      7
32     REAL TMAT ,ENERGY,POWER,SWR                    COMJ      8
33     LOGICAL FTIME                                COMJ      9
34     REAL DB                                        PPWR      5
35     WRITE(*,20) FREQ                              PPWR      6
36     20 FORMAT (/ " POWER BY DIRECTION 1-6 AT A FREQUENCY OF",1PE10.3," PPWR      7
37     +HZ")                                           PPWR      8
38     WRITE (*,30) (POWER(I),I=1,6)                  PPWR      9
39     30 FORMAT ( " 1 2 3 4 5 PPWR     10
40     + 6", /, " ***** PPWR     11
41     +*****", /, 6(3X,F7.2), /, ) PPWR     12
42     WRITE (*,40) PPWR     13
43     40 FORMAT(" ROOM ENERGY DB ", /, PPWR     14
44     + "*****") PPWR     15
45     DO 100 ICOL=1,NROOMS PPWR     16
46     DB=10.0 * ALOG10 ( ENERGY( ICOL) / 10. ) PPWR     17
47     100 WRITE(*,50) ICOL, ENERGY (ICOL), DB PPWR     18
48     50 FORMAT( 3X,I3,5X,F10.2,5X,F10.2) PPWR     19
49     RETURN PPWR     20
50     END PPWR     21

```

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
DB	147B		REAL	
ENERGY	620B	/MAT/	REAL	20
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTIME	652B	/MAT/	LOGICAL	
FTOT	67B	/INITILN/	INTEGER	

I	150B		INTEGER	
ICOL	151B		INTEGER	
IDIR	1043B	/MAT/	INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
POWER	644B	/MAT/	REAL	6
QUALITY	1B	/INITILN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
SWR	653B	/MAT/	REAL	120
TMAT	0B	/MAT/	REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
RMAX	INTEGER	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

ALOG10	REAL	1	INTRINSIC
--------	------	---	-----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

20	55B	FORMAT	36
30	65B	FORMAT	39
40	105B	FORMAT	43
50	117B	FORMAT	48
100	INACTIVE	DO-TERM	47

--ENTRY POINTS--(LO=A)

-NAME-----ADDRESS--ARGS---

PPWR	5B	0
------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	155B = 109
CM LABELLED COMMON LENGTH	2426B = 1302
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.075 SECONDS

```

1      SUBROUTINE PTMAT                                PTMAT      1
2 C** PRINTOUT THE CONTENTS OF THE ROOM MATRIX        PTMAT      2
3 *****COMF                                         1
4 *** COMMON FOR INITIAL PARAMETERS                  ***COMF      2
5 *****COMF                                         3
6      INTEGER FMAX                                    COMF      4
7      PARAMETER (FMAX = 50)                          COMF      5
8      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
9      $          FTOT                                COMF      7
10     COMMON /INITILC/ BLDG                            COMF      8
11     CHARACTER * 5 BLDG                                COMF      9
12     REAL FREQ, AFLAG, RFLAG, FREQA                    COMF     10
13     INTEGER QUALITY, FERR, FTOT                        COMF     11
14 *****COMF                                         12
15 *****COMF                                         13
16 *****COMR                                         1
17 *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
18 *****COMR                                         3
19     INTEGER RMAX                                    COMR      4
20     PARAMETER (RMAX = 20)                          COMR      5
21     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
22     INTEGER NROOMS                                    COMR      7
23     REAL ROOM                                          COMR      8
24 *****COMR                                         9
25 *****COMR                                         10
26 *****COMJ                                         1
27 *                                                    COMJ      2
28 * COMMON FOR EVALUATION OF ROOM MATRIX              COMJ      3
29 *                                                    COMJ      4
30 *****COMJ                                         5
31     COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME COMJ      6
32     +,SWR(RMAX,6),IDIR                                COMJ      7
33     REAL TMAT ,ENERGY,POWER,SWR                      COMJ      8
34     LOGICAL FTIME                                    COMJ      9
35     INTEGER R,C                                       PTMAT      6
36     PRINT*                                           PTMAT      7
37     PRINT*, '      TMAT MATRIX VALUES '            PTMAT      8
38     PRINT*, ' AT FREQUENCY = ',FREQ,' HERTZ'         PTMAT      9
39     PRINT*, '*****'                                PTMAT     10
40     +*****'                                           PTMAT     11
41     DO 10 R = 1,NROOMS                                PTMAT     12
42     PRINT 100,(TMAT(R,C), C = 1, NROOMS )            PTMAT     13
43 10 CONTINUE                                           PTMAT     14
44     PRINT*, '=====PTMAT      15
45     +=====PTMAT      16
46 100 FORMAT(1X,12(E12.6) )                            PTMAT     17
47     RETURN                                           PTMAT     18
48     END                                              PTMAT     19

```

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	146B		INTEGER	
ENERGY	620B	/MAT/	REAL	20
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTIME	652B	/MAT/	LOGICAL	
FTOT	67B	/INITILN/	INTEGER	
IDIR	1043B	/MAT/	INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	

FTN 5.1+552 84/03/14. 10.18.23 PAGE 69
 SUBROUTINE PTMAT 74/175 OPT=0

POWER	644B	/MAT/	REAL	6
QUALITY	1B	/INITILN/	INTEGER	
R	145B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
SWR	653B	/MAT/	REAL	120
TMAT	0B	/MAT/	REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

NAME	TYPE	VALUE
FMAX	INTEGER	50
RMAX	INTEGER	20

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	43
100	113B	FORMAT	46

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
PTMAT	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	153B = 107
CM LABELLED COMMON LENGTH	2426B = 1302
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.068 SECONDS

1	SUBROUTINE DFACTOR	DFACTOR	1
2	*****	COMF	1
3	*** COMMON FOR INITIAL PARAMETERS	***COMF	2
4	*****	COMF	3
5	INTEGER FMAX	COMF	4
6	PARAMETER (FMAX = 50)	COMF	5
7	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,	COMF	6
8	5 FTOT	COMF	7
9	COMMON /INITILC/ BLDG	COMF	8
10	CHARACTER * 5 BLDG	COMF	9
11	REAL FREQ, AFLAG, RFLAG, FREQA	COMF	10
12	INTEGER QUALITY, FERR, FTOT	COMF	11
13	*****	COMF	12
14	*****	COMF	13
15	*****	COMH	1
16	*** COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS	***COMH	2
17	*****	COMH	3
18	INTEGER HMAX	COMH	4
19	PARAMETER (HMAX = 35)	COMH	5
20	COMMON /HOLEN/ HTOT, HERR	COMH	6
21	COMMON /HOLEC/ HOLE(HMAX,4)	COMH	7
22	INTEGER HTOT, HERR	COMH	8
23	CHARACTER * 3 HOLE	COMH	9
24	* =====	COMH	10
25	* DESCRIPTION OF ARRAYS	COMH	11
26	* =====	COMH	12
27	* ROOM IDENTIFICATION APERTURE ID	COMH	13
28	* -----	COMH	14
29	* DIRECTION FROM ROOM TO ROOM	COMH	15
30	* -----	COMH	16
31	* HOLE(X,1) HOLE(X,2) HOLE(X,3) HOLE(X,4)	COMH	17
32	* A3 A3 A3 A3	COMH	18
33	*****	COMH	19
34	*****	COMH	20
35	*****	COMT	1
36	*** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS	***COMT	2
37	*****	COMT	3
38	INTEGER TMAX	COMT	4
39	PARAMETER (TMAX=35)	COMT	5
40	COMMON /TYPEN/ TDIM(TMAX,4), TTOT, TDB2(TMAX,2), TDBTOT, TERR	COMT	6
41	COMMON /TYPEC/ TYPE(TMAX,3), TDB1(TMAX)	COMT	7
42	INTEGER TTOT, TDBTOT, TERR	COMT	8
43	REAL TDIM, TDB2	COMT	9
44	CHARACTER * 3 TYPE, TDB1	COMT	10
45	* =====	COMT	11
46	* DESCRIPTION OF ARRAYS	COMT	12
47	* =====	COMT	13
48	* ID MATERIAL FRAME MATERIAL	COMT	14
49	* -----	COMT	15
50	* TYPE(X,1) TYPE(X,2) TYPE(X,3)	COMT	16
51	* A3 A3 A3	COMT	17
52	* =====	COMT	18
53	* HEIGHT WIDTH LAYER DISTANCE	COMT	19
54	* THICKNESS ABOVE FLOOR	COMT	20
55	* -----	COMT	21
56	* TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4)	COMT	22
57	* F8.2 F8.2 F8.2 F8.2	COMT	23
58	* =====	COMT	24
59	* ID ATTENUATION AREA	COMT	25
60	* -----	COMT	26
61	* TDB1(X) TDB2(X,1) TDB2(X,2)	COMT	27
62	* A3 E9.3 E9.3	COMT	28
63	*****	COMT	29
64	*****	COMT	30

```

65 *****COMR 1
66 *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR 2
67 *****COMR 3
68 INTEGER RMAX COMR 4
69 PARAMETER (RMAX = 20) COMR 5
70 COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR 6
71 INTEGER NROOMS COMR 7
72 REAL ROOM COMR 8
73 *****COMR 9
74 *****COMR 10
75 *****COMW 1
76 *** COMMON FOR DATABASE OF WALL PARAMETERS ***COMW 2
77 *****COMW 3
78 INTEGER WMAX COMW 4
79 PARAMETER (WMAX = 75) COMW 5
80 COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR COMW 6
81 COMMON /WALLC/ WALL(WMAX,4) COMW 7
82 INTEGER WTOT,WERR COMW 8
83 REAL WDIM COMW 9
84 CHARACTER *3 WALL COMW 10
85 * ===== COMW 11
86 ** DESCRIPTION OF ARRAYS COMW 12
87 * ===== COMW 13
88 * WALL IDENTIFICATION COMW 14
89 * ----- COMW 15
90 * DIRECTION FROM TO COMW 16
91 * ROOM ROOM COMW 17
92 * ----- COMW 18
93 * WALL(X,1) WALL(X,2) WALL(X,3) COMW 19
94 * A3 A3 A3 COMW 20
95 * ===== COMW 21
96 * WALL PARAMETERS COMW 22
97 * ----- COMW 23
98 * MATERIAL HEIGHT WIDTH LAYER THICKNESS COMW 24
99 * ----- COMW 25
100 * WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3) COMW 26
101 * A3 F8.2 F8.2 F8.2 COMW 27
102 *****COMW 28
103 *****COMW 29
104 *****COMD 1
105 * COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS COMD 2
106 *****COMD 3
107 COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6), DREFL, DREFLW COMD 4
108 REAL DDABS, DREFL, DREFLW COMD 5
109 *****COMD 6
110 *****COMD 7
111 INTEGER NEXT, LAST, R, ROW DFACTOR 8
112 REAL AFACTOR,DREFLT DFACTOR 9
113 REAL WATTEN,LATTEN,OATTEN,MATTEN,ATTEN,T,S,TS,TS2 DFACTOR 10
114 REAL HEIGHT,WIDTH,AREA,OAREA,WAREA DFACTOR 11
115 CHARACTER *3 FROM, TO, MAT, ID DFACTOR 12
116 LOGICAL NEWWALL,WALLEND DFACTOR 13
117 *** DFACTOR 14
118 *** THIS ROUTINE CALCULATES THE ABSORPTION OF THE WALL AND EACH DFACTOR 15
119 *** OPENING IN THE WALL, LAYER BY LAYER, AND THEN CALCULATES THE DFACTOR 16
120 *** COMPOSITE ABSORPTION BY WEIGHTING BY AREA EACH OPENING'S ABSORPTION DFACTOR 17
121 *** AND THE WALL ABSORPTION. DFACTOR 18
122 *** FOR REFLECTIONS, THE ABSORPTION IS DECREASED BY "DREFL" IF DFACTOR 19
123 *** THE WALL HAS A REFLECTION COEFFICIENT GREATER THAN 0.80 AND DFACTOR 20
124 *** THE ROOM HAS A RESONANCE. DFACTOR 21
125 *** DFACTOR 22
126 *** VARIABLE DEFINITIONS: DFACTOR 23
127 *** DREFLT: EQUALS DREFL IF RESONANCE, OTHERWISE ZERO DFACTOR 24
128 *** IT REPRESENTS REFLECTION GAINS DFACTOR 25

```

129 ***	WATTEN: WALL ATTENUATION	DFACTOR	26
130 ***	OATTEN: OPENING ATTENUATION	DFACTOR	27
131 ***	LATTEN: LAYER ATTENUATION	DFACTOR	28
132 ***	MATTEN: MATERIAL ATTENUATION	DFACTOR	29
133 ***	MAT: MATERIAL IDENTIFICATION	DFACTOR	30
134 ***	WALL: WALL ARRAY CONTAINING WALL IDENTIFICATION AND MATERIAL	DFACTOR	31
135 ***	WDIM: WALL ARRAY CONTAINING PHYSICAL DIMENSIONS OF THE WALL	DFACTOR	32
136 ***	WMAX: MAXIMUM SIZE OF WALL AND WDIM ARRAYS	DFACTOR	33
137 ***	WTOT: TOTAL LINES OF DATA IN THE THE WALL AND WDIM ARRAYS.	DFACTOR	34
138 ***	HEIGHT: HEIGHT OF WALL	DFACTOR	35
139 ***	WIDTH: WIDTH OF WALL	DFACTOR	36
140 ***	T: THICKNESS OF WALL	DFACTOR	37
141 ***	AREA: AREA	DFACTOR	38
142 ***	WAREA: TOTAL WALL AREA WITHOUT SUBTRACTING OPENINGS.	DFACTOR	39
143 ***	OAREA: TOTAL AREA OF THE OPENINGS.	DFACTOR	40
144 ***	NEWWALL: TRUE IF DATA LINE BELONGS TO A NEW WALL	DFACTOR	41
145 ***	WALLEND: TRUE IF DATA LINE IS THE LAST DATA LINE OF A WALL	DFACTOR	42
146 ***		DFACTOR	43
147	DO 10 R = 1,WTOT	DFACTOR	44
148 ***		DFACTOR	45
149 ***	SET WALLEND CONDITION	DFACTOR	46
150	NEXT = R + 1	DFACTOR	47
151	IF (R .EQ. WTOT) THEN	DFACTOR	48
152	WALLEND = .TRUE.	DFACTOR	49
153	ELSE IF (WALL(R,2) .NE. WALL(NEXT,2) .OR.	DFACTOR	50
154	Z WALL(R,3) .NE. WALL(NEXT,3)) THEN	DFACTOR	51
155	WALLEND = .TRUE.	DFACTOR	52
156	ELSE	DFACTOR	53
157	WALLEND = .FALSE.	DFACTOR	54
158	END IF	DFACTOR	55
159 ***		DFACTOR	56
160 ***	SET NEWWALL CONDITION	DFACTOR	57
161	LAST = R - 1	DFACTOR	58
162	IF (R .EQ. 1) THEN	DFACTOR	59
163	NEWWALL = .TRUE.	DFACTOR	60
164	ELSE IF (WALL(R,2) .NE. WALL(LAST,2) .OR.	DFACTOR	61
165	Z WALL(R,3) .NE. WALL(LAST,3)) THEN	DFACTOR	62
166	NEWWALL = .TRUE.	DFACTOR	63
167	ELSE	DFACTOR	64
168	NEWWALL = .FALSE.	DFACTOR	65
169	END IF	DFACTOR	66
170 ***		DFACTOR	67
171 ***	CALCULATE	DFACTOR	68
172	IF (NEWWALL) THEN	DFACTOR	69
173 C*	..INITIALIZE WALL CONDITIONS	DFACTOR	70
174	DREFLT = 0.0	DFACTOR	71
175	TS = 0	DFACTOR	72
176	TS2 = 0	DFACTOR	73
177	WATTEN = 0	DFACTOR	74
178	END IF	DFACTOR	75
179 ***	..CALCULATE ATTENUATION FACTOR OF LAYER	DFACTOR	76
180	MAT = WALL(R,4)	DFACTOR	77
181	MATTEN = ATTEN (MAT,FREQ,AFLAG)	DFACTOR	78
182	CALL RESONW (WALL(R,2) , MAT)	DFACTOR	79
183	IF(DREFLT.GT.0.0) DREFLT = DREFL	DFACTOR	80
184	LATTEN = MATTEN * WDIM(R,3)	DFACTOR	81
185 ***	..CALCULATE RUNNING AFACTOR OF WALL	DFACTOR	82
186	WATTEN = WATTEN + LATTEN	DFACTOR	83
187	IF (WALLEND) THEN	DFACTOR	84
188	FROM = WALL(R,2)	DFACTOR	85
189	TO = WALL(R,3)	DFACTOR	86
190 ***	...CALCULATE WEIGHTED AFACTOR OF OPENINGS	DFACTOR	87
191 ***	...AND TOTAL AREA OF OPENINGS	DFACTOR	88
192	OAREA = 0	DFACTOR	89

193	DO 20 ROW = 1, HTOT	DFACTOR	90
194	IF (HOLE(ROW,2) .EQ. FROM .AND. HOLE(ROW,3) .EQ. TO) THEN	DFACTOR	91
195	ID = HOLE(ROW,4)	DFACTOR	92
196	CALL SRCHTDB(ID, OATTEN, AREA)	DFACTOR	93
197	OAREA = OAREA + AREA	DFACTOR	94
198	IF (OATTEN .LE. 120) THEN	DFACTOR	95
199	T = 1.0 - 10**(-OATTEN / 10)	DFACTOR	96
200	ELSE	DFACTOR	97
201	T = 1.0	DFACTOR	98
202	ENDIF	DFACTOR	99
203	S = AREA	DFACTOR	100
204	TS = TS + T * S	DFACTOR	101
205	TS2 = TS2 + T * S * S	DFACTOR	102
206	END IF	DFACTOR	103
207 20	CONTINUE	DFACTOR	104
208 ***	... CALCULATE TOTAL WALL AREA	DFACTOR	105
209	HEIGHT = WDIM(R,1)	DFACTOR	106
210	WIDTH = WDIM(R,2)	DFACTOR	107
211	WAREA = HEIGHT * WIDTH	DFACTOR	108
212	S = WAREA - OAREA	DFACTOR	109
213	IF (WATTEN .LE. 120.) THEN	DFACTOR	110
214	T = 1.0 - 10**(-WATTEN / 10) -DREFLT	DFACTOR	111
215	ELSE	DFACTOR	112
216	T = 1.0 - DREFLT	DFACTOR	113
217	ENDIF	DFACTOR	114
218	IF(T.LT.0.0) T=0.0	DFACTOR	115
219 ***	... CALCULATE COMPOSITE ATTENUATION FACTOR OF WALL	DFACTOR	116
220	TS = TS + T * S	DFACTOR	117
221	TS2 = TS2 + T * S * S	DFACTOR	118
222 ***	... INSERT COMPOSITE ATTENUATION OF WALL INTO ROOM MATRIX	DFACTOR	119
223	CALL LDDABS (TS,TS2,FROM,TO)	DFACTOR	120
224	END IF	DFACTOR	121
225 10	CONTINUE	DFACTOR	122
226	RETURN	DFACTOR	123
227	END	DFACTOR	124

--VARIABLE MAP--(LO=A)

--NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

AFACTOR	NONE	UNUSED/*S*	REAL	
AFLAG	2B	/INITILN/	REAL	
AREA	446B		REAL	
BLDG	0B	/INITILC/	CHAR*5	
DDABS	0B	/ROOMD/	REAL	676
DREFL	1244B	/ROOMD/	REAL	
DREFLT	433B		REAL	
DREFLW	1245B	/ROOMD/	REAL	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FROM	451B		CHAR*3	
FTOT	67B	/INITILN/	INTEGER	
HEIGHT	444B		REAL	
HERR	1B	/HOLEN/	INTEGER	
HOLE	0B	/HOLEC/	CHAR*3	140
HTOT	0B	/HOLEN/	INTEGER	
ID	454B		CHAR*3	
LAST	430B		INTEGER	
LATTEN	435B		REAL	
MAT	453B		CHAR*3	
MATTEN	437B		REAL	
NEWWALL	455B		LOGICAL	
NEXT	427B		INTEGER	

NROOMS	1244B	/ROOMN/	INTEGER	
OAREA	447B		REAL	
OATTEN	436B		REAL	
QUALITY	1B	/INITILN/	INTEGER	
R	431B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
ROW	432B		INTEGER	
S	441B		REAL	
T	440B		REAL	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPECL/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TO	452B		CHAR*3	
TS	442B		REAL	
TS2	443B		REAL	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPECL/	CHAR*3	105
WALL	0B	/WALLC/	CHAR*3	300
WALLEND	456B		LOGICAL	
WAREA	450B		REAL	
WATTEN	434B		REAL	
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WIDTH	445B		REAL	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

NAME	TYPE	VALUE
FMAX	INTEGER	50
HMAX	INTEGER	35
RMAX	INTEGER	20
TMAX	INTEGER	35
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ATTEN	REAL	3	FUNCTION
LDDABS		4	SUBROUTINE
RESONW		2	SUBROUTINE
SRCHTDB		3	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	225
20	INACTIVE	DO-TERM	207

FTN 5.1+552 84/03/14. 10.18.23 PAGE 75
SUBROUTINE DFACTOR 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

DFACTOR 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH	463B = 307
CM LABELLED COMMON LENGTH	4000B = 2048
CM STORAGE USED	63000B = 26112
COMPILE TIME	0.230 SECONDS

```

1      SUBROUTINE LDDABS ( TS, TS2, FROM, TO )                                LDDABS      1
2      *****LDDABS                                                         LDDABS      2
3      *   THIS ROUTINE LOADS THE ABSORPTION COEFFICIENT INTO THE APPROPRIATE LDDABS      3
4      *   LOCATION IN THE 'DDABS' ARRAY.                                     LDDABS      4
5      *                                                                 LDDABS      5
6      *   NROOMS: TOTAL NUMBERS OF ROOMS REPRESENTED BY DATA                LDDABS      6
7      *   RMAX: MAXIMUM NUMBER POSSIBLE UNDER THE PRESENT PROGRAM CONFIGURATIO LDDABS      7
8      *   TS AND TS2: ABSORPTION COEFFICIENTS                                LDDABS      8
9      *   FROM: TO: CONTAINS ROOM#'S OR THE DIRECTIONS D1,D2,4,D5,OR D6.     LDDABS      9
10     *****LDDABS                                                         LDDABS     10
11     *****COMR                                                            COMR       1
12     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS                ***COMR     2
13     *****COMR                                                            COMR       3
14     INTEGER RMAX                                                            COMR       4
15     PARAMETER (RMAX = 20)                                                  COMR       5
16     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)          COMR       6
17     INTEGER NROOMS                                                         COMR       7
18     REAL ROOM                                                             COMR       8
19     *****COMR                                                            COMR       9
20     *****COMR                                                            COMR      10
21     *****COMD                                                            COMD       1
22     *   COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS          COMD       2
23     *****COMD                                                            COMD       3
24     COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6), DREFL, DREFLW                COMD       4
25     REAL DDABS, DREFL, DREFLW                                              COMD       5
26     *****COMD                                                            COMD       6
27     *****COMD                                                            COMD       7
28     INTEGER VAL, C, R, RNUM, D                                            LDDABS     13
29     REAL TS, TS2                                                            LDDABS     14
30     CHARACTER * 3 FROM, TO                                                LDDABS     15
31     *****LDDABS                                                         LDDABS     16
32     IF ( FROM(1:1) .EQ. 'D' ) THEN                                       LDDABS     17
33         RNUM = VAL ( TO(1:2) )                                           LDDABS     18
34         D = VAL ( FROM(2:2) )                                           LDDABS     19
35     *****LDDABS                                                         LDDABS     20
36     *   INSERT ABSORPTION COEFFICIENT FOR ENERGY ENTERING A ROOM FROM THE LDDABS     21
37     *   OUTSIDE OF THE BUILDING.                                          LDDABS     22
38     *****LDDABS                                                         LDDABS     23
39         R = NROOMS + D                                                    LDDABS     24
40         C = RNUM                                                            LDDABS     25
41         DDABS(R,C) = TS + DDABS(R,C)                                     LDDABS     26
42     *****LDDABS                                                         LDDABS     27
43     *   INSERT ABSORPTION COEFFICIENT INTO 'DDABS' ARRAY FOR ENERGY LEAVING LDDABS     28
44     *   A ROOM TO THE OUTSIDE OF THE BUILDING.                          LDDABS     29
45     *****LDDABS                                                         LDDABS     30
46         R = RNUM                                                            LDDABS     31
47         C = NROOMS + D                                                    LDDABS     32
48         DDABS(R,C) = TS2 / RAREA(RNUM) + DDABS(R,C)                    LDDABS     33
49     *****LDDABS                                                         LDDABS     34
50     ELSE IF ( TO(1:1) .EQ. 'D' ) THEN                                    LDDABS     35
51         RNUM = VAL ( FROM(1:2) )                                         LDDABS     36
52         D = VAL ( TO(2:2) )                                              LDDABS     37
53     *****LDDABS                                                         LDDABS     38
54     *   INSERT ABSORPTION COEFFICIENT INTO 'DDABS' ARRAY FOR ENERGY ENTERING LDDABS     39
55     *   A ROOM FROM THE OUTSIDE OF THE BUILDING.                        LDDABS     40
56     *****LDDABS                                                         LDDABS     41
57         R = NROOMS + D                                                    LDDABS     42
58         C = RNUM                                                            LDDABS     43
59         DDABS(R,C) = TS + DDABS(R,C)                                     LDDABS     44
60     *****LDDABS                                                         LDDABS     45
61     *   INSERT ABSORPTION COEFFICIENT INTO 'DDABS' ARRAY FOR ENERGY LEAVING LDDABS     46
62     *   A ROOM TO THE OUTSIDE OF THE BUILDING.                          LDDABS     47
63     *****LDDABS                                                         LDDABS     48
64         R = RNUM                                                            LDDABS     49

```

```

65          C = NROOMS + D                                LDDABS      50
66          DDABS(R,C) = TS2 / RAREA(RNUM) + DDABS(R,C)    LDDABS      51
67 *****LDDABS      52
68          ELSE                                           LDDABS      53
69 *****LDDABS      54
70 *    INSERT ABSORPTION COEFFICIENTS INTO 'DDABS' ARRAY FOR ENERGY GOING LDDABS      55
71 *    FROM ROOM TO ROOM.                                LDDABS      56
72 *****LDDABS      57
73          R = VAL ( FROM(1:2) )                          LDDABS      58
74          C = VAL ( TO(1:2) )                            LDDABS      59
75          DDABS(R,C) = TS2 / RAREA(R) + DDABS(R,C)        LDDABS      60
76          DDABS(C,R) = TS2 / RAREA(C) + DDABS(C,R)        LDDABS      61
77          ENDIF                                           LDDABS      62
78          RETURN                                           LDDABS      63
79          END                                              LDDABS      64

```

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	222B		INTEGER	
D	225B		INTEGER	
DDABS	0B	/ROOMD/	REAL	676
DREPL	1244B	/ROOMD/	REAL	
DREPLW	1245B	/ROOMD/	REAL	
FROM	3	DUMMY-ARG	CHAR*3	
NROOMS	1244B	/ROOMN/	INTEGER	
R	223B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RNUM	224B		INTEGER	
ROOM	0B	/ROOMN/	REAL	676
TO	4	DUMMY-ARG	CHAR*3	
TS	1	DUMMY-ARG	REAL	
TS2	2	DUMMY-ARG	REAL	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

RMAX	INTEGER	20
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

VAL	INTEGER	1	FUNCTION
-----	---------	---	----------

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

LDDABS	5B	4
--------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	230B = 152
CM LABELLED COMMON LENGTH	2537B = 1375
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.103 SECONDS

```

1      SUBROUTINE PDDABS                                PDDABS      1
2 C** PRINTOUT THE CONTENTS OF THE ROOM MATRIX          PDDABS      2
3 *****COMF                                           1
4 *** COMMON FOR INITIAL PARAMETERS                    ***COMF      2
5 *****COMF                                           3
6      INTEGER FMAX                                      COMF        4
7      PARAMETER (FMAX = 50)                            COMF        5
8      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF        6
9      $          FTOT                                    COMF        7
10     COMMON /INITILC/ BLDG                             COMF        8
11     CHARACTER * 5 BLDG                                COMF        9
12     REAL FREQ, AFLAG, RFLAG, FREQA                     COMF       10
13     INTEGER QUALITY, FERR, FTOT                        COMF       11
14 *****COMF                                           12
15 *****COMF                                           13
16 *****COMR                                           1
17 *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
18 *****COMR                                           3
19     INTEGER RMAX                                       COMR        4
20     PARAMETER (RMAX = 20)                             COMR        5
21     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR        6
22     INTEGER NROOMS                                    COMR        7
23     REAL ROOM                                          COMR        8
24 *****COMR                                           9
25 *****COMR                                           10
26 *****COMJ                                           1
27 *                                                     COMJ        2
28 * COMMON FOR EVALUATION OF ROOM MATRIX                COMJ        3
29 *                                                     COMJ        4
30 *****COMJ                                           5
31     COMMON /MAT/ TMAT(RMAX, RMAX), ENERGY(RMAX), POWER(6), FTIME COMJ        6
32     +, SWR(RMAX, 6), IDIR                             COMJ        7
33     REAL TMAT, ENERGY, POWER, SWR                   COMJ        8
34     LOGICAL FTIME                                    COMJ        9
35 *****COMD                                           1
36 * COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS COMD        2
37 *****COMD                                           3
38     COMMON /ROOMD/ DDABS(RMAX + 6, RMAX + 6), DREFL, DREFLW      COMD        4
39     REAL DDABS, DREFL, DREFLW                         COMD        5
40 *****COMD                                           6
41 *****COMD                                           7
42     INTEGER R, C                                      PDDABS      7
43     PRINT*                                             PDDABS      8
44     PRINT*, ' DDABS MATRIX VALUES '                  PDDABS      9
45     PRINT*, ' AT FREQUENCY = ', FREQ, ' HERTZ'         PDDABS     10
46     PRINT*, ' WITH AFLAG = ', AFLAG, ' PER CENT'      PDDABS     11
47     PRINT*, '*****'                                  PDDABS     12
48     +*****'                                           PDDABS     13
49     DO 10 R = 1, NROOMS + 6                            PDDABS     14
50     PRINT 100, (DDABS(R, C), C = 1, NROOMS + 6 )     PDDABS     15
51 10 CONTINUE                                           PDDABS     16
52     PRINT*, '===== PDDABS     17
53     +===== PDDABS     18
54 100 FORMAT(1X, 12(F8.3) ) PDDABS     19
55     RETURN                                           PDDABS     20
56     END                                             PDDABS     21

```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
AFLAG	2B	/INITILN/		REAL	
BLDG	0B	/INITILC/		CHAR*5	
C	163B			INTEGER	
DDABS	0B	/ROOMD/		REAL	676
DREFL	1244B	/ROOMD/		REAL	
DREFLW	1245B	/ROOMD/		REAL	
ENERGY	620B	/MAT/		REAL	20
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FTIME	652B	/MAT/		LOGICAL	
FTOT	67B	/INITILN/		INTEGER	
IDIR	1043B	/MAT/		INTEGER	
NROOMS	1244B	/ROOMN/		INTEGER	
POWER	644B	/MAT/		REAL	6
QUALITY	1B	/INITILN/		INTEGER	
R	162B			INTEGER	
RAREA	1245B	/ROOMN/		REAL	20
RFLAG	3B	/INITILN/		REAL	
ROOM	0B	/ROOMN/		REAL	676
SWR	653B	/MAT/		REAL	120
TMAT	0B	/MAT/		REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

NAME	TYPE	VALUE
FMAX	INTEGER	50
RMAX	INTEGER	20

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	51
100	121B	FORMAT	54

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
PDDABS	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	170B = 120
CM LABELLED COMMON LENGTH	3674B = 1980
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.071 SECONDS


```

1      SUBROUTINE 1DDABS                                IDDABS      1
2      *****IDDABS                                IDDABS      2
3      *  INITIALIZE DDABS MATRIX.                      IDDABS      3
4      *****IDDABS                                IDDABS      4
5      *****COMR                                COMR            1
6      ***  COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS  ***COMR      2
7      *****COMR                                COMR            3
8      INTEGER RMAX                                COMR            4
9      PARAMETER (RMAX = 20)                        COMR            5
10     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)  COMR            6
11     INTEGER NROOMS                                COMR            7
12     REAL ROOM                                     COMR            8
13     *****COMR                                COMR            9
14     *****COMR                                COMR           10
15     *****COMD                                COMD            1
16     *  COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS  COMD            2
17     *****COMD                                COMD            3
18     COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ), DREFL, DREFLW  COMD            4
19     REAL DDABS ,DREFL , DREFLW                    COMD            5
20     *****COMD                                COMD            6
21     *****COMD                                COMD            7
22     INTEGER R,C                                IDDABS            7
23     DO 10 R = 1,RMAX                            IDDABS            8
24     DO 10 C = 1,RMAX                            IDDABS            9
25     DDABS(R,C) = 0.0                            IDDABS           10
26 10  CONTINUE                                    IDDABS           11
27     DO 20 R = 1,RMAX                            IDDABS           12
28     DO 20 C = RMAX + 1, RMAX + 5                IDDABS           13
29     DDABS(R,C) = 0.0                            IDDABS           14
30 20  CONTINUE                                    IDDABS           15
31     RETURN                                       IDDABS           16
32     END                                         IDDABS           17

```

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

C	70B		INTEGER	
DDABS	0B	/ROOMD/	REAL	676
DREFL	1244B	/ROOMD/	REAL	
DREFLW	1245B	/ROOMD/	REAL	
NROOMS	1244B	/ROOMN/	INTEGER	
R	67B		INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
ROOM	0B	/ROOMN/	REAL	676

--SYMBOLIC CONSTANTS--(LO=A)

-NAME---TYPE-----VALUE

RMAX	INTEGER	20
------	---------	----

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

10	INACTIVE	DO-TERM	26
20	INACTIVE	DO-TERM	30

FTN 5.1+552 84/03/14. 10.18.23 PAGE 81
SUBROUTINE IDDABS 74/175 OPT=0

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

IDDABS 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH	77B = 63
CM LABELLED COMMON LENGTH	2537B = 1375
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.050 SECONDS

```
1      SUBROUTINE PPWR2                                PPWR2      1
2      *****COMF                                     1
3      *** COMMON FOR INITIAL PARAMETERS                ***COMF      2
4      *****COMF                                     3
5      INTEGER FMAX                                     COMF      4
6      PARAMETER (FMAX = 50)                            COMF      5
7      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
8      $          FTOT                                    COMF      7
9      COMMON /INITILC/ BLDG                            COMF      8
10     CHARACTER * 5 BLDG                                COMF      9
11     REAL FREQ, AFLAG, RFLAG, FREQA                     COMF     10
12     INTEGER QUALITY, FERR, FTOT                        COMF     11
13     *****COMF                                     12
14     *****COMF                                     13
15     *****COMR                                     1
16     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
17     *****COMR                                     3
18     INTEGER RMAX                                       COMR      4
19     PARAMETER (RMAX = 20)                            COMR      5
20     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
21     INTEGER NROOMS                                    COMR      7
22     REAL ROOM                                          COMR      8
23     *****COMR                                     9
24     *****COMR                                    10
25     *****COMJ                                     1
26     *                                                  COMJ      2
27     * COMMON FOR EVALUATION OF ROOM MATRIX            COMJ      3
28     *                                                  COMJ      4
29     *****COMJ                                     5
30     COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME COMJ      6
31     +,SWR(RMAX,6),IDIR                                COMJ      7
32     REAL TMAT,ENERGY,POWER,SWR                       COMJ      8
33     LOGICAL FTIME                                    COMJ      9
34     REAL DB                                           PPWR2     5
35     WRITE(*,30) FREQ                                  PPWR2     6
36     30 FORMAT (//,"ATTENUATION AT A FREQUENCY OF",1PE10.3," HZ" PPWR2     7
37     +,/, "*****", PPWR2     8
38     + "*****", PPWR2     9
39     +,/, " * * * * * DIRECTIONS ",28X,"*",/, PPWR2    10
40     + " * ROOMS * 1 2 3 4 " , PPWR2    11
41     + " 5 *",/, "*****", PPWR2    12
42     + "*****") PPWR2    13
43     WRITE(*,100) (IROW,(SWR(IROW,I),I=1,5), IROW=1,NROOMS) PPWR2    14
44     100 FORMAT (" * ",I3,3X,"*",F9.2,3X,F10.2,3X,F10.2,3X,F10.2, PPWR2    15
45     + " *") PPWR2    16
46     WRITE (*,120) PPWR2    17
47     120 FORMAT ("*****", PPWR2    18
48     + "*****") PPWR2    19
49     RETURN PPWR2    20
50     END PPWR2    21
```

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/ INITILN/	REAL	
BLDG	0B	/ INITILC/	CHAR*5-	
DB	NONE	UNUSED/*S*	REAL	
ENERGY	620B	/MAT/	REAL	20
FERR	66B	/ INITILN/	INTEGER	
FREQ	0B	/ INITILN/	REAL	
FREQA	4B	/ INITILN/	REAL	50
FTIME	652B	/MAT/	LOGICAL	
FTOT	67B	/ INITILN/	INTEGER	

I	167B		INTEGER	
IDIR	1043B	/MAT/	INTEGER	
IROW	165B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
POWER	644B	/MAT/	REAL	6
QUALITY	1B	/INITILN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
ROOM	0B	/ROOMN/	REAL	676
SWR	653B	/MAT/	REAL	120
TMAT	0B	/MAT/	REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
RMAX	INTEGER	20

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES----DEF

30	56B	FORMAT	36
100	123B	FORMAT	44
120	133B	FORMAT	47

--ENTRY POINTS--(LO=A)

-NAME-----ADDRESS--ARGS---

FPWR2	5B	0
-------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	173B = 123
CM LABELLED COMMON LENGTH	2426B = 1302
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.071 SECONDS

```

1      SUBROUTINE SPWR                                SPWR      1
2      *****COMF                                  COMF      1
3      *** COMMON FOR INITIAL PARAMETERS                ***COMF      2
4      *****COMF                                  COMF      3
5      INTEGER FMAX                                COMF      4
6      PARAMETER (FMAX = 50)                        COMF      5
7      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
8      $      FTOT                                COMF      7
9      COMMON /INITILC/ BLDG                        COMF      8
10     CHARACTER * 5 BLDG                            COMF      9
11     REAL FREQ, AFLAG, RFLAG, FREQA                COMF     10
12     INTEGER QUALITY, FERR, FTOT                    COMF     11
13     *****COMF                                  COMF     12
14     *****COMF                                  COMF     13
15     *****COMR                                  COMR      1
16     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
17     *****COMR                                  COMR      3
18     INTEGER RMAX                                COMR      4
19     PARAMETER (RMAX = 20)                        COMR      5
20     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
21     INTEGER NROOMS                                COMR      7
22     REAL ROOM                                    COMR      8
23     *****COMR                                  COMR      9
24     *****COMR                                  COMR     10
25     *****COMJ                                  COMJ      1
26     *                                              COMJ      2
27     * COMMON FOR EVALUATION OF ROOM MATRIX          COMJ      3
28     *                                              COMJ      4
29     *****COMJ                                  COMJ      5
30     COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME COMJ      6
31     +,SWR(RMAX,6),IDIR                            COMJ      7
32     REAL TMAT ,ENERGY,POWER,SWR                    COMJ      8
33     LOGICAL FTIME                                COMJ      9
34     REAL DB                                        SPWR      5
35     DO 100 IROW=1,NROOMS                          SPWR      6
36     IF(ENERGY(IROW).LT.1.0E-05) THEN                SPWR      7
37     DB = -60.0                                     SPWR      8
38     GO TO 100                                       SPWR      9
39     ENDIF                                           SPWR     10
40     DB=10.0 * ALOG10 ( ENERGY( IROW) / 10. )      SPWR     11
41     100 SWR(IROW,IDIR)=DB                          SPWR     12
42     RETURN                                          SPWR     13
43     END                                            SPWR     14

```

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
DB	55B		REAL	
ENERGY	620B	/MAT/	REAL	20
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTIME	652B	/MAT/	LOGICAL	
FTOT	67B	/INITILN/	INTEGER	
IDIR	1043B	/MAT/	INTEGER	
IROW	56B		INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
POWER	644B	/MAT/	REAL	6
QUALITY	1B	/INITILN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	

FTN 5.1+552 84/03/14. 10.18.23 PAGE 85
 SUBROUTINE SPWR 74/175 OPT=0

ROOM	0B	/ROOMN/	REAL	676
SWR	653B	/MAT/	REAL	120
TMAT	0B	/MAT/	REAL	400

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
RMAX	INTEGER	20

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

ALOG10	REAL	1	INTRINSIC
--------	------	---	-----------

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

100	31B	DO-TERM	41
-----	-----	---------	----

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS---ARGS---

SPWR	5B	0
------	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH	62B = 50
CM LABELLED COMMON LENGTH	2426B = 1302
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.057 SECONDS

```

1      SUBROUTINE RESONW(FROM,MATID)                                RESONW    1
2      *                                                            RESONW    2
3      *      THIS ROUTINE CHECKS FOR RESONANCE CONDITIONS IN A ROOM RESONW    3
4      *      IF A RESONANCE IS POSSIBLE AT THE PARTICULAR FREQUENCY RESONW    4
5      *      FOR THE ROOM AND THE WALL IN QUESTION HAS A REFLECTION RESONW    5
6      *      COEFFICIENT GREATER THAN 0.80, THEN THE ABSORPTION FOR RESONW    6
7      *      THE WALL IS REDUCED BY THE REFLECTION COEFFICIENT. RESONW    7
8      *                                                            RESONW    8
9      *****COMR 1
10     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR 2
11     *****COMR 3
12     INTEGER RMAX COMR 4
13     PARAMETER (RMAX = 20) COMR 5
14     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR 6
15     INTEGER NROOMS COMR 7
16     REAL ROOM COMR 8
17     *****COMR 9
18     *****COMR 10
19     *****COMW 1
20     *** COMMON FOR DATABASE OF WALL PARAMETERS ***COMW 2
21     *****COMW 3
22     INTEGER WMAX COMW 4
23     PARAMETER (WMAX = 75) COMW 5
24     COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR COMW 6
25     COMMON /WALLC/ WALL(WMAX,4) COMW 7
26     INTEGER WTOT,WERR COMW 8
27     REAL WDIM COMW 9
28     CHARACTER *3 WALL COMW 10
29     * ===== COMW 11
30     ** DESCRIPTION OF ARRAYS COMW 12
31     * ===== COMW 13
32     * WALL IDENTIFICATION COMW 14
33     * ----- COMW 15
34     * DIRECTION FROM TO COMW 16
35     * ROOM ROOM COMW 17
36     * ----- COMW 18
37     * WALL(X,1) WALL(X,2) WALL(X,3) COMW 19
38     * A3 A3 A3 COMW 20
39     * ===== COMW 21
40     * WALL PARAMETERS COMW 22
41     * ----- COMW 23
42     * MATERIAL HEIGHT WIDTH LAYER THICKNESS COMW 24
43     * ----- COMW 25
44     * WALL(X,4) WDIM(X,1) WDIM(X,2) WDIM(X,3) COMW 26
45     * A3 F8.2 F8.2 F8.2 COMW 27
46     *****COMW 28
47     *****COMW 29
48     *****COMF 1
49     *** COMMON FOR INITIAL PARAMETERS ***COMF 2
50     *****COMF 3
51     INTEGER FMAX COMF 4
52     PARAMETER (FMAX = 50) COMF 5
53     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF 6
54     $ FTOT COMF 7
55     COMMON /INITILC/ BLDG COMF 8
56     CHARACTER * 5 BLDG COMF 9
57     REAL FREQ, AFLAG, RFLAG, FREQA COMF 10
58     INTEGER QUALITY, FERR, FTOT COMF 11
59     *****COMF 12
60     *****COMF 13
61     *****COMM 1
62     *** COMMON FOR DATABASE OF MATERIAL PROPERTIES ***COMM 2
63     *****COMM 3
64     INTEGER MMAX COMM 4

```

65	PARAMETER (MMAX=100)	COMM	5
66	COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX),	COMM	6
67	\$ MFREQ(MMAX,7), MERR, MTOT	COMM	7
68	COMMON /MATC/MAT(MMAX),MATDESC(MMAX)	COMM	8
69	INTEGER MTOT, MERR	COMM	9
70	REAL MATTEN, MRCOEF, MFREQ, QA, QR	COMM	10
71	CHARACTER * 3 MAT	COMM	11
72	CHARACTER * 70 MATDESC	COMM	12
73	*****	COMM	13
74	*****	COMM	14
75	*****	COMM	1
76	* COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS	COMM	2
77	*****	COMM	3
78	COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6), DREFL, DREFLW	COMM	4
79	REAL DDABS, DREFL, DREFLW	COMM	5
80	*****	COMM	6
81	*****	COMM	7
82	REAL A,B,C,RH,RL,RW	RESONW	14
83	REAL MREFL	RESONW	15
84	CHARACTER * 3 FROM,MATID	RESONW	16
85	ISSET = 0	RESONW	17
86	MREFL = RCOEF (MATID,FREQ,RFLAG)	RESONW	18
87	IF(MREFL.LT.0.80) THEN	RESONW	19
88	DREFL = 0.0	RESONW	20
89	RETURN	RESONW	21
90	ENDIF	RESONW	22
91	IF(FROM(1:1) .EQ. 'D') THEN	RESONW	23
92	DREFL = 0.0	RESONW	24
93	RETURN	RESONW	25
94	ENDIF	RESONW	26
95	*	RESONW	27
96	NOW GET HEIGHT, WIDTH AND LENGTH OF ROOM	RESONW	28
97	*	RESONW	29
98	DO 100 I1 = 1, WTOT	RESONW	30
99	IF(FROM.EQ.WALL(I1,2) .AND. WALL(I1,1).EQ.'FB ') THEN	RESONW	31
100	RH = WDIM(I1,1)	RESONW	32
101	GOTO 200	RESONW	33
102	ENDIF	RESONW	34
103	100 CONTINUE	RESONW	35
104	DREFL = 0	RESONW	36
105	IWARN = 9	RESONW	37
106	CALL WARNING(IWARN)	RESONW	38
107	RETURN	RESONW	39
108	200 I1 = 0	RESONW	40
109	300 I1 = I1 + 1	RESONW	41
110	IF(I1.GT.WTOT) THEN	RESONW	42
111	DREFL = 0.0	RESONW	43
112	IF(ISSET.EQ.1) RETURN	RESONW	44
113	IWARN = 10	RESONW	45
114	CALL WARNING(IWARN)	RESONW	46
115	RETURN	RESONW	47
116	ENDIF	RESONW	48
117	IF(FROM.EQ.WALL(I1,2) .AND. WALL(I1,1).EQ.'UD ') THEN	RESONW	49
118	RL= WDIM(I1,2)	RESONW	50
119	RW= WDIM(I1,1)	RESONW	51
120	ELSE	RESONW	52
121	GOTO 300	RESONW	53
122	ENDIF	RESONW	54
123	*	RESONW	55
124	NOW SORT OUT DIMENSIONS WITH A SMALLEST AND C LARGEST	RESONW	56
125	ISSET = 1	RESONW	57
126	IPASS = 0	RESONW	58
127	A = RH	RESONW	59
128	B = RW	RESONW	60

129	C = RL	RESONW	61
130	500 IF (A.GT.B) THEN	RESONW	62
131	TMP= A	RESONW	63
132	A = B	RESONW	64
133	B = TMP	RESONW	65
134	IPASS = 0	RESONW	66
135	ELSE	RESONW	67
136	IPASS = 1	RESONW	68
137	ENDIF	RESONW	69
138	IF (B.GT.C) THEN	RESONW	70
139	TMP =B	RESONW	71
140	B = C	RESONW	72
141	C = TMP	RESONW	73
142	IPASS = 0	RESONW	74
143	ENDIF	RESONW	75
144	IF (IPASS.EQ. 0) GOTO 500	RESONW	76
145 *		RESONW	77
146 *	NOW CALCULATE LOWER RESONANCE FREQUENCY	RESONW	78
147	CLIGHT= 3.0E08	RESONW	79
148	FLOW = 1.0/(B*B) + 1.0/ (C*C)	RESONW	80
149	FLOW = SQRT(FLOW)	RESONW	81
150	FLOW = FLOW*CLIGHT/2.0	RESONW	82
151 *		RESONW	83
152 *	NOW CALCULATE HIGH FREQUENCY LIMIT	RESONW	84
153	FHIGH = 9.0*(1.0/(A*A) + 1.0/(B*B) + 1.0/(C*C))	RESONW	85
154	FHIGH = SQRT (FHIGH)	RESONW	86
155	FHIGH = FHIGH*CLIGHT/2.0	RESONW	87
156	IF(FREQ.GE.FLOW .AND. FREQ.LE.FHIGH) THEN	RESONW	88
157	DREFL = MREFL	RESONW	89
158	RETURN	RESONW	90
159	ELSE	RESONW	91
160	DREFL = 0.0	RESONW	92
161	GOTO 300	RESONW	93
162	ENDIF	RESONW	94
163	END	RESONW	95

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

A	346B		REAL	
AFLAG	2B	/INITILN/	REAL	
B	347B		REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	350B		REAL	
CLIGHT	363B		REAL	
DDABS	0B	/ROOMD/	REAL	676
DREFL	1244B	/ROOMD/	REAL	
DREFLW	1245B	/ROOMD/	REAL	
FERR	66B	/INITILN/	INTEGER	
FHIGH	365B		REAL	
FLOW	364B		REAL	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FROM	1	DUMMY-ARG	CHAR*3	
FTOT	67B	/INITILN/	INTEGER	
IPASS	361B		INTEGER	
ISSET	355B		INTEGER	
IWARN	360B		INTEGER	
I1	356B		INTEGER	
MAT	0B	/MATC/	CHAR*3	100
MATDESC	36B	/MATC/	CHAR*70	100
MATID	2	DUMMY-ARG	CHAR*3	
MATTEN	0B	/MATN/	REAL	700

MERR	4374B	/MATN/	INTEGER	
MFREQ	3100B	/MATN/	REAL	700
MRCOEF	1274B	/MATN/	REAL	700
MREFL	354B		REAL	
MTOT	4375B	/MATN/	INTEGER	
NROOMS	1244B	/ROOMN/	INTEGER	
QA	2570B	/MATN/	REAL	100
QR	2734B	/MATN/	REAL	100
QUALITY	1B	/INITILN/	INTEGER	
RAREA	1245B	/ROOMN/	REAL	20
RFLAG	3B	/INITILN/	REAL	
RH	351B		REAL	
RL	352B		REAL	
ROOM	0B	/ROOMN/	REAL	676
RW	353B		REAL	
TMP	362B		REAL	
WALL	0B	/WALLC/	CHAR*3	300
WDIM	0B	/WALLN/	REAL	225
WERR	342B	/WALLN/	INTEGER	
WTOT	341B	/WALLN/	INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

NAME	TYPE	VALUE
FMAX	INTEGER	50
MMAX	INTEGER	100
RMAX	INTEGER	20
WMAX	INTEGER	75

--PROCEDURES--(LO=A)

NAME	TYPE	ARCS	CLASS
RCOEF	REAL	3	FUNCTION
SQRT	GENERIC	1	INTRINSIC
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
100	INACTIVE	DO-TERM	103
200	102B		108
300	104B		109
500	162B		130

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARCS
RESONW	5B	2

--STATISTICS--

PROGRAM-UNIT LENGTH	370B = 248
CM LABELLED COMMON LENGTH	11255B = 4781
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.190 SECONDS


```

1      SUBROUTINE RESOND(ID)                                RESOND      1
2      *                                                    RESOND      2
3      *                                                    RESOND      3
4      *      THIS ROUTINE CALCULATES RESONANCES FOR DOORS AND RESOND      4
5      *      WINDOWS. IN THE RESONANCE FREQUENCY RANGE, THE RESOND      5
6      *      INPUT TRANSMISSION OF THE WINDOW OR DOOR IS IN- RESOND      6
7      *      BY 20 DB (ARBITRARY).                          RESOND      7
8      *                                                    RESOND      8
9      *                                                    RESOND      9
10     *****COMR                                           1
11     *** COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS ***COMR      2
12     *****COMR                                           3
13     INTEGER RMAX                                           COMR      4
14     PARAMETER (RMAX = 20)                                  COMR      5
15     COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX) COMR      6
16     INTEGER NROOMS                                         COMR      7
17     REAL ROOM                                              COMR      8
18     *****COMR                                           9
19     *****COMR                                           10
20     *****COMD                                           1
21     * COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS COMD      2
22     *****COMD                                           3
23     COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6), DREFL, DREFLW COMD      4
24     REAL DDABS, DREFL, DREFLW                             COMD      5
25     *****COMD                                           6
26     *****COMD                                           7
27     *****COMF                                           1
28     *** COMMON FOR INITIAL PARAMETERS                      ***COMF      2
29     *****COMF                                           3
30     INTEGER FMAX                                           COMF      4
31     PARAMETER (FMAX = 50)                                  COMF      5
32     COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF      6
33     $          FTOT                                         COMF      7
34     COMMON /INITILC/ BLDG                                  COMF      8
35     CHARACTER * 5 BLDG                                     COMF      9
36     REAL FREQ, AFLAG, RFLAG, FREQA                        COMF     10
37     INTEGER QUALITY, FERR, FTOT                           COMF     11
38     *****COMF                                           12
39     *****COMF                                           13
40     *****COMT                                           1
41     *** COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS ***COMT      2
42     *****COMT                                           3
43     INTEGER TMAX                                           COMT      4
44     PARAMETER (TMAX=35)                                    COMT      5
45     COMMON /TYPEN/ TDIM(TMAX,4), TTOT, TDB2(TMAX,2), TDBTOT, TERR COMT      6
46     COMMON /TYPEC/ TYPE(TMAX,3), TDB1(TMAX)               COMT      7
47     INTEGER TTOT, TDBTOT, TERR                            COMT      8
48     REAL TDIM, TDB2                                       COMT      9
49     CHARACTER * 3 TYPE, TDB1                              COMT     10
50     *=====COMT                                           11
51     * DESCRIPTION OF ARRAYS                                COMT     12
52     *=====COMT                                           13
53     * ID MATERIAL FRAME MATERIAL                          COMT     14
54     *-----COMT                                           15
55     *TYPE(X,1) TYPE(X,2) TYPE(X,3)                        COMT     16
56     * A3 A3 A3                                             COMT     17
57     *-----COMT                                           18
58     * HEIGHT WIDTH LAYER DISTANCE                         COMT     19
59     * THICKNESS ABOVE FLOOR                               COMT     20
60     *-----COMT                                           21
61     * TDIM(X,1) TDIM(X,2) TDIM(X,3) TDIM(X,4)            COMT     22
62     * F8.2 F8.2 F8.2 F8.2                                COMT     23
63     *-----COMT                                           24
64     * ID ATTENUATION AREA                                  COMT     25

```

65	*	-----	COMT	26
66	*	TDB1(X) TDB2(X,1) TDB2(X,2)	COMT	27
67	*	A3 E9.3 E9.3	COMT	28
68	*****			COMT
69	*****			COMT
70		CHARACTER * 3, ID, MATID	RESOND	14
71		DO 10 I = 1, TTOT	RESOND	15
72		J=I	RESOND	16
73		IF(TYPE(I,1) .EQ. ID) GOTO 20	RESOND	17
74	10	CONTINUE	RESOND	18
75		IWARN = 11	RESOND	19
76		CALL WARNING(IWARN)	RESOND	20
77		DREFLW = 0.0	RESOND	21
78		RETURN	RESOND	22
79	20	CONTINUE	RESOND	23
80		MATID = TYPE (J,3)	RESOND	24
81		RH = TDIM (J,1)	RESOND	25
82		RW = TDIM (J,2)	RESOND	26
83		REFL = RCOEF(MATID, FREQ, RFLAG)	RESOND	27
84		IF(REFL .LT. 0.80) THEN	RESOND	28
85		DREFLW = 0.0	RESOND	29
86		RETURN	RESOND	30
87		ENDIF	RESOND	31
88		A = RH	RESOND	32
89		B = RW	RESOND	33
90		IF(A.GT.B) THEN	RESOND	34
91		TMP = B	RESOND	35
92		B = A	RESOND	36
93		A = TMP	RESOND	37
94		ENDIF	RESOND	38
95		FLOW = 3.0E8 / 2/ B	RESOND	39
96		FHIGH = 3.0E8 / 2.0 * 3.0 * SQRT(1/(B*B) + 1 / (A*A))	RESOND	40
97		IF(FREQ.GE.FLOW .AND. FREQ.LE.FHIGH) THEN	RESOND	41
98		DREFLW = 20.0	RESOND	42
99		ELSE	RESOND	43
100		DREFLW = 0.0	RESOND	44
101		ENDIF	RESOND	45
102		RETURN	RESOND	46
103		END	RESOND	47

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS---BLOCK-----PROPERTIES-----TYPE-----SIZE

A	205B		REAL	
AFLAG	2B	/INITILN/	REAL	
B	206B		REAL	
BLDG	0B	/INITILC/	CHAR*5	
DDABS	0B	/ROOMD/	REAL	676
DREFL	1244B	/ROOMD/	REAL	
DREFLW	1245B	/ROOMD/	REAL	
FERR	66B	/INITILN/	INTEGER	
FHIGH	211B		REAL	
FLOW	210B		REAL	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
I	176B		INTEGER	
ID	1	DUMMY-ARG	CHAR*3	
IWARN	201B		INTEGER	
J	200B		INTEGER	
MATID	175B		CHAR*3	
NROOMS	1244B	/ROOMN/	INTEGER	
QUALITY	1B	/INITILN/	INTEGER	

RARCA	1245B	/ROOMN/	REAL	20
REFL	204B		REAL	
RFLAG	3B	/INITILN/	REAL	
RH	202B		REAL	
ROOM	0B	/ROOMN/	REAL	676
RW	203B		REAL	
TDBTOT	323B	/TYPEN/	INTEGER	
TDB1	37B	/TYPEC/	CHAR*3	35
TDB2	215B	/TYPEN/	REAL	70
TDIM	0B	/TYPEN/	REAL	140
TERR	324B	/TYPEN/	INTEGER	
TMP	207B		REAL	
TTOT	214B	/TYPEN/	INTEGER	
TYPE	0B	/TYPEC/	CHAR*3	105

--SYMBOLIC CONSTANTS--(LO=A)

NAME	TYPE	VALUE
FMAX	INTEGER	50
RMAX	INTEGER	20
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
RCOEF	REAL	3	FUNCTION
SQRT	GENERIC	1	INTRINSIC
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	74
20	43B		79

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
RESOND	5B	1

--STATISTICS--

PROGRAM-UNIT LENGTH	214B = 140
CM LABELLED COMMON LENGTH	3227B = 1687
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.120 SECONDS

1	SUBROUTINE LFREQ		LFREQ	1
2	*[[[LFREQ		LFREQ	2
3	*[[[LFREQ		LFREQ	3
4	*[[[LOAD THE CONTENTS OF THE FILE 'BXXXXXF' INTO ARRAYS FREQA.		LFREQ	4
5	*[[[LFREQ		LFREQ	5
6	*[[[LFREQ		LFREQ	6
7	*****LFREQ		LFREQ	7
8	*****COMF		COMF	1
9	*** COMMON FOR INITIAL PARAMETERS		***COMF	2
10	*****COMF		COMF	3
11	INTEGER FMAX		COMF	4
12	PARAMETER (FMAX = 50)		COMF	5
13	COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,		COMF	6
14	\$ FTOT		COMF	7
15	COMMON /INITILC/ BLDG		COMF	8
16	CHARACTER * 5 BLDG		COMF	9
17	REAL FREQ, AFLAG, RFLAG, FREQA		COMF	10
18	INTEGER QUALITY, FERR, FTOT		COMF	11
19	*****COMF		COMF	12
20	*****COMF		COMF	13
21	*****LFREQ		LFREQ	9
22	* DECLARATION OF VARIABLES		LFREQ	10
23	*****LFREQ		LFREQ	11
24	INTEGER GETLEN, R, C		LFREQ	12
25	CHARACTER * 7 NAME, PFN		LFREQ	13
26	*****LFREQ		LFREQ	14
27	*		LFREQ	15
28	*****LFREQ		LFREQ	16
29	NAME = 'B'//BLDG(1:GETLEN(BLDG))//'F'		LFREQ	17
30	PFN = NAME (1:GETLEN(NAME))		LFREQ	18
31	FERR = 0 :G		LFREQ	19
32	CALL PF ('GET',0,PFN(1:GETLEN(PFN)), 'RC', FERR)		LFREQ	20
33	IF (FERR .EQ. 0) THEN		LFREQ	21
34	OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',		LFREQ	22
35	\$ STATUS='OLD', ACCESS='SEQUENTIAL')		LFREQ	23
36	FTOT = 0		LFREQ	24
37	DO 10 R = 1,FMAX		LFREQ	25
38	READ (3,1000,END=20) FREQA(R)		LFREQ	26
39	1000 FORMAT(E12.7)		LFREQ	27
40	FTOT = FTOT + 1		LFREQ	28
41	10 CONTINUE		LFREQ	29
42	20 CONTINUE		LFREQ	30
43	CLOSE(3,STATUS='DELETE')		LFREQ	31
44	ELSE IF (FERR .EQ. 2) THEN		LFREQ	32
45	CALL WARNING (11)		LFREQ	33
46	ELSE		LFREQ	34
47	CALL WARNING (12)		LFREQ	35
48	END IF		LFREQ	36
49	RETURN		LFREQ	37
50	END		LFREQ	38

--VARIABLE MAP--(LO=A)

-NAME---ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE

AFLAG	2B	/INITILN/	REAL	
BLDG	0B	/INITILC/	CHAR*5	
C	NONE	UNUSED/*S*	INTEGER	
FERR	66B	/INITILN/	INTEGER	
FREQ	0B	/INITILN/	REAL	
FREQA	4B	/INITILN/	REAL	50
FTOT	67B	/INITILN/	INTEGER	
NAME	210B		CHAR*7	
PFN	211B		CHAR*7	

FTN 5 1+552 84/03/14. 10.18.23 PAGE 94
SUBROUTINE LFREQ 74/175 OPT=0

QUALITY	1B	/INITILN/	INTEGER
R	207B		INTEGER
RFLAG	3B	/INITILN/	REAL

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-----TYPE-----VALUE

FMAX	INTEGER	50
------	---------	----

--PROCEDURES--(LO=A)

-NAME-----TYPE-----ARGS-----CLASS-----

GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-ADDRESS-----PROPERTIES-----DEF

10	INACTIVE	DO-TERM	41
20	63B		42
1000	120B	FORMAT	39

--ENTRY POINTS--(LO=A)

-NAME---ADDRESS--ARGS---

LFREQ	5B	0
-------	----	---

--I/O UNITS--(LO=A)

-NAME--- PROPERTIES-----

TAPE3	AUX/FMT/SEQ
-------	-------------

--STATISTICS--

PROGRAM-UNIT LENGTH	215B = 141
CM LABELLED COMMON LENGTH	71B = 57
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.085 SECONDS

Appendix 9.8 Blank Forms for Data Taking.

WALLS DATA FORM

BUILDING I. D. NUMBER _____ DATE _____

NAME _____

[illegible]

HOLES DATA FORM

BUILDING I.D. NUMBER _____

DATE _____

NAME _____

[illegible]

TYPES DATA FORM
(for windows and doors)

BUILDING I.D. NUMBER _____ DATE _____

NAME _____

[illegible]

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 84-3009	2. Performing Organ. Report No.	3. Publication Date September 1984
4. TITLE AND SUBTITLE BUILDING PENETRATION PROJECT			
5. AUTHOR(S) J. C. Wyss, W. J. Anson, R. D. Orr			
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			7. Contract/Grant No. 8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP) Communications Electronics Engineering Installation Agency Fort Huachuca, Arizona 08613			
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FiPS Software Summary, is attached.			
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here) This report documents a computer program which calculates building attenuation of electromagnetic radiation over the frequency range 10 kHz - 10 GHz. Attenuation (in dB) is computed from building shape, dimensions, room layout, and the electrical properties of construction materials; no electromagnetic measurements are required. Details of the structure and use of the program are given.			
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) computer model; electromagnetic attenuation; electromagnetic interference; electromagnetic shielding; shielding materials			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 310 15. Price \$25.00

